HP ProCurve Switch Software

IPv6 Configuration Guide

3500yl switches
5400zl switches
6200yl switches
6600 switches
8212zl switch

Software version: K.14.24
June 2009

HP ProCurve

3500yl Switches
5400zl Switches
6200yl Switch
6600 Switches
8212zl Switch

June 2009
K.14.24

IPv6 Configuration Guide

© Copyright 2008 - 2009 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. All Rights Reserved.

This document contains proprietary information, which is protected by copyright. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard.

Publication Number

5992-3067
June 2009

Applicable Products

HP ProCurve Switch 3500yl-24G-PWR	(J8692A)
HP ProCurve Switch 3500yl-48G-PWR	(J8693A)
HP ProCurve Switch 5406zl	(J8697A)
HP ProCurve Switch 5412zl	(J8698A)
HP ProCurve Switch 6200yl-24G	(J8992A)
HP ProCurve Switch 8212zl	(J8715A)
HP ProCurve Switch 6600-24G	(J9263A)
HP ProCurve Switch 6600-24G-4XG	(J9264A)
HP ProCurve Switch 6600-24G-24XG	(J9265A)
HP ProCurve Switch 6600-48G	(J9451A)
HP ProCurve Switch 6600-48G-4XG	(J9452A)

Trademark Credits

Microsoft, Windows, and Microsoft Windows NT are US registered trademarks of Microsoft Corporation. Java™ is a US trademark of Sun Microsystems, Inc.

Disclaimer

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD COMPANY MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Warranty

See the Customer Support/Warranty booklet included with the product.

A copy of the specific warranty terms applicable to your Hewlett-Packard products and replacement parts can be obtained from your HP Sales and Service Office or authorized dealer.

Hewlett-Packard Company
8000 Foothills Boulevard, m/s 5551
Roseville, California 95747-5551
http://www.procurve.com

Contents

Product Publications and IPv6 Command Index

About Your Switch Manual Set xiii

Printed Publications. xiii

Electronic Publications xiii

IPv6 Command Index xv

1 Getting Started

Contents 1-1

Introduction 1-2

Conventions 1-2

Command Syntax Statements 1-2

Command Prompts 1-3

Screen Simulations 1-3

Configuration and Operation Examples 1-3

Keys 1-4

Sources for More Information 1-4

Getting Documentation From the Web 1-7

Online Help 1-7

Menu Interface 1-7

Command Line Interface 1-8

Web Browser Interface 1-8

To Set Up and Install the Switch in Your Network 1-9

2 Introduction to IPv6

Contents 2-1

Migrating to IPv6 2-3

IPv6 Propagation 2-4

Dual-Stack Operation 2-4

Connecting to Devices Supporting IPv6 Over IPv4 Tunneling 2-5

Information Sources for Tunneling IPv6 Over IPv4 2-5
Use Model 2-6
Adding IPv6 Capability 2-6
Supported IPv6 Operation 2-6
Configuration and Management 2-7
Management Features 2-7
IPv6 Addressing 2-7
SLAAC (Stateless Automatic Address Configuration) 2-7
DHCPv6 (Stateful) Address Configuration 2-8
Static Address Configuration 2-8
Default IPv6 Gateway 2-8
Neighbor Discovery (ND) in IPv6 2-9
IPv6 Management Features 2-10
TFTPv6 Transfers 2-10
IPv6 Time Configuration 2-10
Telnet6 2-10
IP Preserve 2-11
Multicast Listener Discovery (MLD) 2-11
Web Browser Interface 2-11
Path MTU (PMTU) Discovery 2-11
Configurable IPv6 Security 2-12
SSHv2 on IPv6 2-12
IP Authorized Managers 2-12
Diagnostic and Troubleshooting 2-14
ICMP Rate-Limiting 2-14
Ping6 2-14
Traceroute6 2-14
Debug/Syslog Enhancements 2-14
Domain Name System (DNS) Resolution 2-14
IPv6 Neighbor Discovery (ND) Controls 2-15
Event Log 2-15
SNMP 2-15
Loopback Address 2-15
IPv6 Scalability 2-16

3 IPv6 Addressing

Contents 3-1

Introduction 3-3

IPv6 Address Structure and Format 3-3

Address Format 3-3

Address Notation 3-3

Network Prefix 3-4

Interface (Device) Identifier 3-4

IPv6 Addressing Options 3-5

IPv6 Address Sources 3-5

General IPv6 Address Types 3-5

IPv6 Address Sources 3-7

Stateless Address Autoconfiguration (SLAAC) 3-7

Applications 3-7

Preferred and Valid Lifetimes of Stateless Autoconfigured Addresses 3-7

Stateful (DHCPv6) Address Configuration 3-8

Static Address Configuration 3-9

Address Types and Scope 3-10

Address Types 3-10

Address Scope 3-11

Unicast Address Prefixes 3-11

Link-Local Unicast Address 3-13

Autoconfiguring Link-Local Unicast Addresses 3-13

Extended Unique Identifier (EUI) 3-14

Statically Configuring Link-Local Addresses 3-15

Global Unicast Address 3-16

Stateless Autoconfiguration of a Global Unicast Address 3-16

Static Configuration of a Global Unicast Address 3-17

Prefixes in Routable IPv6 Addresses 3-18

Unique Local Unicast IPv6 Address 3-19

Anycast Addresses 3-20

Multicast Application to IPv6 Addressing 3-21

Overview of the Multicast Operation in IPv6 3-21
IPv6 Multicast Address Format 3-22
Multicast Group Identification 3-22
Solicited-Node Multicast Address Format 3-23
Loopback Address 3-24
The Unspecified Address 3-25
IPv6 Address Deprecation 3-25
Preferred and Valid Address Lifetimes 3-25

4 IPv6 Addressing Configuration

Contents 4-1
Introduction 4-3
General Configuration Steps 4-4
Configuring IPv6 Addressing 4-5
Enabling IPv6 with an Automatically Configured Link-Local Address 4-6
Enabling Autoconfiguration of a Global Unicast Address and a Default Router Identity on a VLAN 4-7
Operating Notes 4-8
Enabling DHCPv6 4-9
Operating Notes 4-10
Configuring a Static IPv6 Address on a VLAN 4-11
Statically Configuring a Link-Local Unicast Address 4-12
Statically Configuring A Global Unicast Address 4-13
Operating Notes 4-14
Statically Configuring An Anycast Address 4-14
Duplicate Address Detection (DAD) for Statically Configured Addresses 4-16
Disabling IPv6 on a VLAN 4-16
Neighbor Discovery (ND) 4-17
Duplicate Address Detection (DAD) 4-18
DAD Operation 4-18
Configuring DAD 4-19

Operating Notes for Neighbor Discovery 4-20
View the Current IPv6 Addressing Configuration 4-22
Router Access and Default Router Selection 4-29
Router Advertisements 4-29
Router Solicitations 4-29
Default IPv6 Router 4-30
Router Redirection 4-30
View IPv6 Gateway, Route, and Router Neighbors 4-31
Viewing Gateway and IPv6 Route Information 4-31
Viewing IPv6 Router Information 4-32
Address Lifetimes 4-34
Preferred Lifetime 4-34
Valid Lifetime 4-34
Sources of IPv6 Address Lifetimes 4-34

5 IPv6 Management Features

Contents 5-1
Introduction 5-2
Viewing and Clearing the IPv6 Neighbors Cache 5-2
Viewing the Neighbor Cache 5-3
Clearing the Neighbor Cache 5-5
IPv6 Telnet Operation 5-6
Outbound Telnet to Another Device 5-6
Viewing the Current Telnet Activity on a Switch 5-7
Enabling or Disabling Inbound Telnet Access 5-8
Viewing the Current Inbound Telnet Configuration 5-8
SNTP and Timep 5-9
Configuring (Enabling or Disabling) the SNTP Mode 5-9
Configuring an IPv6 Address for an SNTP Server 5-10
Configuring (Enabling or Disabling) the Timep Mode 5-12
TFTP File Transfers Over IPv6 5-15
Enabling TFTP for IPv6 5-16
Using TFTP to Copy Files over IPv6 5-17

Using Auto-TFTP for IPv6 5-20
SNMP Management for IPv6 5-21
SNMP Features Supported 5-21
SNMP Configuration Commands Supported 5-22
SNMPv1 and V2c 5-22
SNMPv3 5-22
IP Preserve for IPv6 5-24

6 IPv6 Management Security Features

Contents 6-1
IPv6 Management Security 6-2
Authorized IP Managers for IPv6 6-3
Usage Notes 6-3
Configuring Authorized IP Managers for Switch Access 6-5
Using a Mask to Configure Authorized Management Stations 6-5
Configuring Single Station Access 6-5
Configuring Multiple Station Access 6-6
Displaying an Authorized IP Managers Configuration 6-12
Additional Examples of Authorized IPv6 Managers Configuration . 6-13
Secure Shell (SSH) for IPv6 6-15
Configuring SSH for IPv6 6-15
Displaying an SSH Configuration 6-18
Secure Copy and Secure FTP for IPv6 6-19

7 Multicast Listener Discovery (MLD) Snooping

Contents 7-1
Overview 7-2
Introduction to MLD Snooping 7-3
Configuring MLD 7-8
Enabling or Disabling MLD Snooping on a VLAN 7-8
Configuring Per-Port MLD Traffic Filters 7-9
Configuring the Querier 7-10
Configuring Fast Leave 7-10

Configuring Forced Fast Leave 7-11
Displaying MLD Status and Configuration 7-12
Current MLD Status 7-12
Current MLD Configuration 7-15
Ports Currently Joined 7-17
Statistics 7-18
Counters 7-20

8 IPv6 Access Control Lists (ACLs)

Contents 8-1
Introduction 8-4
Overview of Options for Applying IPv6 ACLs on the Switch 8-6
Static ACLS 8-6
RADIUS-Assigned ACLs 8-6
Command Summary for Configuring ACLs 8-7
Command Summary for Enabling, Disabling, and Displaying ACLs . 8-8
Terminology 8-9
Overview 8-13
Types of IPv6 ACLs 8-13
Concurrent IPv4 and IPv6 ACLs 8-13
IPv6 ACL Applications 8-13
VACL Applications 8-15
IPv6 Static Port ACL Applications 8-16
RADIUS-Assigned (Dynamic) Port ACL Applications 8-16
Multiple ACL Assignments on an Interface 8-18
Features Common to All ACL Applications 8-21
General Steps for Planning and Configuring ACLs 8-22
IPv6 ACL Operation 8-24
Introduction 8-24
The Packet-filtering Process 8-24
Planning an ACL Application 8-28
IPv6 Traffic Management and Improved Network Performance . . . 8-28
Security 8-29
Guidelines for Planning the Structure of an ACL 8-30

ACL Configuration and Operating Rules 8-31
How an ACE Uses a Prefix To Screen Packets for SA and DA Matches 8-33
Prefix Usage Differences Between ACLs and Other IPv6 Addressing 8-34
Configuring and Assigning an IPv6 ACL 8-35
General Steps for Implementing IPv6 ACLs 8-35
Permit/Deny Options 8-36
ACL Configuration 8-36
ACL Configuration Structure 8-38
ACL Configuration Factors 8-40
The Sequence of Entries in an ACL Is Significant 8-40
Allowing for the Implied Deny Function 8-41
A Configured ACL Has No Effect Until You Apply It to an Interface 8-42
You Can Assign an ACL Name to an Interface Even if the ACL Has Not Been Configured 8-42
Using the CLI To Create an ACL 8-42
General ACE Rules 8-43
Using CIDR Notation To Enter the IPv6 ACL Prefix Length . . . 8-43
Configuration Commands 8-45
Command Summary for Configuring ACLs 8-45
Command Summary for Enabling, Disabling, and Displaying ACLs 8-46
Overview 8-46
Commands To Create, Enter, and Configure an ACL 8-47
Adding or Removing an ACL Assignment On an Interface 8-62
Filtering Switched IPv6 Traffic Inbound on a VLAN 8-62
Filtering Inbound IPv6 Traffic Per Port and Trunk 8-63
Deleting an ACL 8-65
Editing an Existing ACL 8-66
General Editing Rules 8-66
Sequence Numbering in ACLs 8-66
Inserting an ACE in an Existing ACL 8-68
Deleting an ACE from an Existing ACL 8-70
Resequencing the ACEs in an IPv6 ACL 8-71

Attaching a Remark to an ACE 8-73
Operating Notes for Remarks 8-76
Displaying ACL Configuration Data 8-78
Display an ACL Summary 8-79
Display the Content of All ACLs on the Switch 8-80
Display the IPv4 and IPv6 VACL Assignments for a VLAN 8-81
Display Static Port (and Trunk) ACL Assignments 8-82
Displaying the Content of a Specific ACL 8-83
Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File 8-86
Creating or Editing ACLs Offline 8-87
Creating or Editing an ACL Offline 8-87
The Offline Process 8-87
Example of Using the Offline Process 8-88
Testing and Troubleshooting ACLs 8-92
Enable IPv6 ACL “Deny” Logging 8-92
Requirements for Using IPv6 ACL Logging 8-92
ACL Logging Operation 8-93
Enabling ACL Logging on the Switch 8-93
Monitoring Static ACL Performance 8-96
Example of ACL Performance Monitoring 8-98
IPv6 Counter Operation with Multiple Interface Assignments . 8-99
IPv4 Counter Operation with Multiple Interface Assignments 8-101
General ACL Operating Notes 8-105

9 IPv6 Diagnostic and Troubleshooting

Contents 9-1
Introduction 9-2
ICMP Rate-Limiting 9-2
Ping for IPv6 (Ping6) 9-4
Traceroute for IPv6 9-6
DNS Resolver for IPv6 9-9
DNS Configuration 9-9

Viewing the Current Configuration 9-11
Operating Notes 9-11
Debug/Syslog for IPv6 9-12
Configuring Debug and Event Log Messaging 9-12
Debug Command 9-13
Configuring Debug Destinations 9-15
Logging Command 9-16

A IPv6 Terminology

Product Publications and IPv6 Command Index

About Your Switch Manual Set

Note

For the latest version of all ProCurve switch documentation, including Release Notes covering recently added features, please visit the ProCurve Networking Web site at **www.procurve.com,** click on **Technical support**, and then click on **Product manuals (all)**.

Printed Publications

The two publications listed below are printed and shipped with your switch. The latest version of each is also available in PDF format on the ProCurve Web site, as described in the above Note.

- *Read Me First*—Provides software update information, product notes, and other information.
- *Installation and Getting Started Guide*—Explains how to prepare for and perform the physical installation and connect the switch to your network.

Electronic Publications

The latest version of each publication listed in this section (including the above printed publications) is available in PDF format on the ProCurve Web site, as described in the Note at the top of this page.

The six publications listed below cover all of the switches supported by this manual.

- *Management and Configuration Guide*—Describes how to configure, manage, and monitor basic switch operation.
- *Advanced Traffic Management Guide*—Explains how to configure traffic management features such as VLANs, MSTP, QoS, and Meshing.
- *Multicast and Routing Guide*—Explains how to configure IGMP, PIM, IP routing, and VRRP features.
- *Access Security Guide*—Explains how to configure access security features and user authentication on the switch.
- *IPv6 Configuration Guide*—Describes the IPv6 protocol operations that are supported on the switch.
- *Release Notes*—Describe new features, fixes, and enhancements that become available between revisions of the main product guide.

The two publications listed below support all of the switches covered by this manual *except* the ProCurve Series 2900 switches:

- *Command Line Interface Reference Guide*—Provides a comprehensive description of CLI commands, syntax, and operations.
- *Event Log Message Reference Guide*—Provides a comprehensive description of event log messages.

IPv6 Command Index

This index provides a tool for locating descriptions of individual IPv6 commands covered in this guide.

Note

A link-local address must include **%vlan< *vid* >** without spaces as a suffix. For example:

```
fe80::110:252%vlan20
```

The index begins on the next page.

Command	Min. Level	Page
Authorized Manager		
ipv6 authorized managers < *ipv6-addr* >*	Global Config	6-5
show ipv6 authorized-managers	Manager	6-12
Copy		
auto-tftp	Global Config	5-20
copy tftp < *target* > < *ipv6-addr* > < *filename* >	Manager	5-17
copy < *source* > tftp < *ipv6-addr* > < *filename* >	Manager	5-18
tftp6 [client | server]	Global Config	5-16
Debug/Syslog		
debug ipv6 < dhcpv6-client | nd >	Manager	9-13
logging < *syslog-ipv4-addr* >	Global Config	9-16
Diagnostic		
ping6	Operator	9-4
traceroute6	Operator	9-6
DNS		
ip dns domain-name < *domain-name-str* >	Global Config	9-10
ip dns server-address priority < 1 - 3 > < *ipv6-addr* >*	Global Config	9-9
IPv6 Addressing		
ipv6 address autoconfig	VLAN Config	4-7
ipv6 address dhcp full [rapid-commit]	VLAN Config	4-9
ipv6 address fe80::< *device-id* > link-local	VLAN Config	4-12
ipv6 address < *ipv6-addr* >/< *prefix-len* >	VLAN Config	4-13
ipv6 address < *ipv6-addr* >/< *prefix-len* > eui-64	VLAN Config	4-13
ipv6 address < *ipv6-addr* >/< *prefix-len* > anycast	VLAN Config	4-15
show ipv6	Operator	4-22
show ipv6 vlan < *vid* >	Operator	4-24
IPv6 Management		
clear ipv6 neighbors	Manager	5-5
ip preserve *(Command file entry; not a CLI command.)*	n/a	5-24
ipv6 enable	VLAN Config	4-6
ipv6 icmp error-interval < 0 - 2147483647 >	Global Config	9-3

*A link-local address in these commands must include **%vlan< *vid* >** as a suffix. For example, `fe80::110:252%vlan20.`

Command	Min. Level	Page
IPv6 Management *(Continued)*		
ipv6 nd dad-attempts < 0 - 600 >	Global Config	4-19
ipv6 nd ns-interval < 1000 - 3600000 >	VLAN Config	4-20
ipv6 nd reachable-time < 1000 - 2147483647 >	VLAN Config	4-20
show ipv6 neighbors	Operator	5-3
show ipv6 nd	Operator	4-25
show ipv6 route	Operator	4-31
show ipv6 routers	Operator	4-32
snmp-server host < *ipv6-addr* >*	Global Config	5-22
MLD		
ipv6 mld	VLAN Config	7-8
ipv6 mld [< auto | blocked | forward > < *port-list* >]	VLAN Config	7-9
ipv6 mld fastleave < *port-list* >	VLAN Config	7-10
ipv6 mld forcedfastleave < *port-list* >	VLAN Config	7-11
ipv6 mld querier	VLAN Config	7-10
show ipv6 mld vlan < *vid* >	Operator	7-12
config	Operator	7-15
group [*ipv6-addr*]*	Operator	7-17
statistics	Operator	7-18
counters	Operator	7-20
SSH		
ip ssh [cipher | filetransfer | mac | port | public-key | timeout]	Global Config	6-19
Telnet		
show console	Operator	5-8
show telnet	Operator	5-7
telnet < *ipv6-addr* >*	Manager	5-6
Timep		
ip timep dhcp	Global Config	5-13
ip timep manual < *ipv6-addr* >*	Global Config	5-13
show sntp	Manager	5-11
show timep	Manager	5-14
sntp server priority < 1 - 3 > < *ipv6-addr* >*	Global Config	5-10

*A link-local address in these commands must include **%vlan< *vid* >** as a suffix. For example, `fe80::110:252%vlan20`.

1

Getting Started

Contents

Introduction 1-2

Conventions 1-2

- Command Syntax Statements 1-2
- Command Prompts 1-3
- Screen Simulations 1-3
- Configuration and Operation Examples 1-3
- Keys 1-4

Sources for More Information 1-4

- Getting Documentation From the Web 1-7
- Online Help 1-7
  - Menu Interface 1-7
  - Command Line Interface 1-8
  - Web Browser Interface 1-8

To Set Up and Install the Switch in Your Network 1-9

Introduction

Beginning with software release K.14.01, this guide is intended for use with the following switches:

- ProCurve Switch 8200zl series
- ProCurve Switch 5400zl series
- ProCurve Switch 3500yl and 6200yl series

It describes how to use the command line interface (CLI) to configure, manage, monitor, and troubleshoot switch operation. For an overview of other product documentation for the above switches, refer to *"Product Documentation"* on page xiii. You can download documentation from the ProCurve Networking web site, **www.procurve.com**.

Conventions

This guide uses the following conventions for command syntax and displayed information.

Command Syntax Statements

Syntax: ip < default-gateway < *ip-addr* >> | routing >

Syntax: show interfaces [*port-list*]

- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate optional elements.
- Braces (< >) enclose required elements.
- Braces within square brackets ([< >]) indicate a required element within an optional choice.
- Boldface indicates use of a CLI command, part of a CLI command syntax, or other displayed element in general text. For example:

  "Use the **copy tftp** command to download the key from a TFTP server."
- Italics indicate variables for which you must supply a value when executing the command. For example, in this command syntax, you must provide one or more port numbers:

  Syntax: telnet < *ipv6-address* >

Command Prompts

In the default configuration, your switch displays a CLI prompt similar to the following example:

```
ProCurve 8212zl#
```

To simplify recognition, this guide uses **ProCurve** to represent command prompts for all switch models. For example:

```
ProCurve#
```

(You can use the **hostname** command to change the text in the CLI prompt.)

Screen Simulations

Displayed Text. Figures containing simulated screen text and command output look like this:

```
ProCurve> show version

Image stamp:    /sw/code/build/info

                January 10 2009 14:28:59

                K.14.01

                314

Boot Image:     Primary

ProCurve>
```

Figure 1-1. Example of a Figure Showing a Simulated Screen

In some cases, brief command-output sequences appear without figure identification. For example:

```
ProCurve(config)# clear public-key
ProCurve(config)# show ip client-public-key
show_client_public_key: cannot stat keyfile
```

Configuration and Operation Examples

Unless otherwise noted, examples using a particular switch model apply to all switch models covered by this guide.

Keys

Simulations of actual keys use a bold, sans-serif typeface with square brackets. For example, the Tab key appears as **[Tab]** and the "Y" key appears as **[Y]**.

Sources for More Information

This guide covers features related to IPv6 operation in software release K.14.01, and includes an IPv6 command index on page xv.

For information about switch operation and features not covered in this guide, refer to the switch publications listed in this section.

Note

For the latest version of all ProCurve switch documentation referred to below, including Release Notes covering recently added features, visit the ProCurve Networking web site at **www.procurve.com**, click on **Technical support**, and then click on **Product Manuals (all)**.

- Software Release Notes—*Release Notes* are posted on the ProCurve Networking web site and provide information on new software updates:
  - new features and how to configure and use them
  - software management, including downloading software to the switch
  - software fixes addressed in current and previous releases
- Product Notes and Software Update Information—The printed *Read Me First* shipped with your switch provides software update information, product notes, and other information.
- *Installation and Getting Started Guide*—Use the *Installation and Getting Started Guide* shipped with your switch to prepare for and perform the physical installation. This guide also steps you through connecting the switch to your network and assigning IP addressing, as well as describing the LED indications for correct operation and trouble analysis.
- *Management and Configuration Guide*—Use this guide for information on topics such as:
  - various interfaces available on the switch
  - memory and configuration operation
  - interface access
  - IP addressing
  - time protocols

  - port configuration, trunking, traffic control, and PoE operation
  - Redundant management
  - SNMP, LLDP, and other network management topics
  - file transfers, switch monitoring, troubleshooting, and MAC address management
- *Advanced Traffic Management Guide*—Use this guide for information on topics such as:
  - VLANs: Static port-based and protocol VLANs, and dynamic GVRP VLANs
  - spanning-Tree: 802.1D (STP), 802.1w (RSTP), and 802.1s (MSTP)
  - meshing
  - Quality-of-Service (QoS)
  - IPv4 Access Control Lists (ACLs)
- *Multicast and Routing Guide*—Use this guide for information on topics such as:
  - IGMP
  - PIM (SM and DM)
  - IP routing
  - VRRP
- *Access Security Guide*—Use this guide for information on topics such as:
  - Local username and password security
  - Web-Based and MAC-based authentication
  - RADIUS and TACACS+ authentication
  - RADIUS-assigned rate-limiting, CoS, and ACLs
  - SSH (Secure Shell) and SSL (Secure Socket Layer) operation
  - 802.1X access control
  - Port security operation with MAC-based control
  - Authorized IP Manager security
  - Key Management System (KMS)
- *IPv6 Configuration Guide*—Use this guide for information on topics such as:
  - Overview of IPv6 operation and supported features
  - Configuring IPv6 addressing
  - Using IPv6 management, security, and troubleshooting features
- Feature Index—The following software guides for your switch include an index of non-IPv6 features (and where to find them). This index immediately precedes the first chapter in each guide listed.

- *Management and Configuration Guide*
- *Advanced Traffic Management Guide*
- *Access Security Guide*
- *Multicast and Routing Guide*

Getting Documentation From the Web

To obtain the latest versions of documentation and release notes for your switch:

1. Go to the ProCurve Networking web site at

   www.procurve.com

2. Click on **Technical support**.
3. Click on **Product manuals**.
4. Click on the product for which you want to view or download a manual.

If you need further information on ProCurve switch technology, visit the ProCurve Networking web site at:

www.procurve.com

Online Help

Menu Interface

If you need information on specific parameters in the menu interface, refer to the online help provided in the interface. For example:

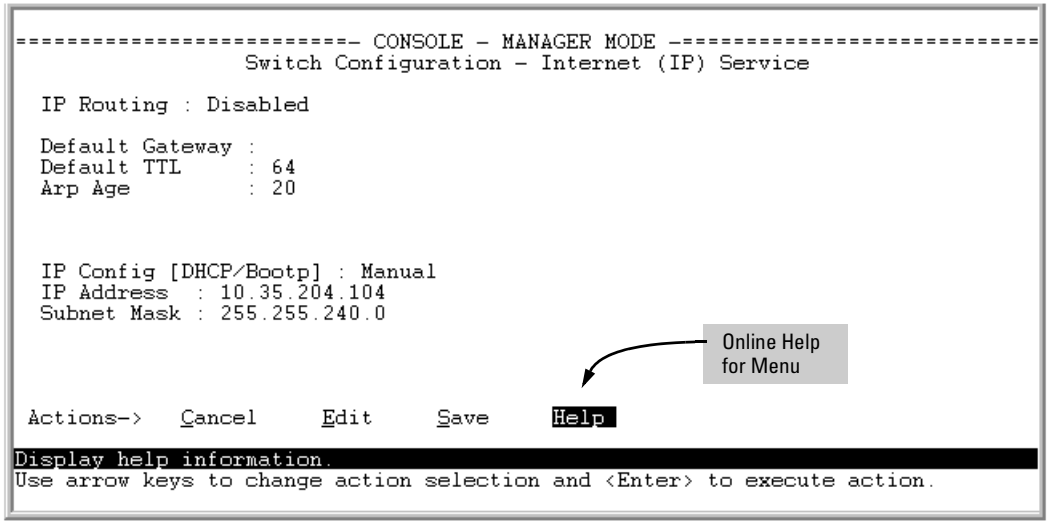
========== CONSOLE - MANAGER MODE ==========
Switch Configuration - Internet (IP) Service
IP Routing : Disabled
Default Gateway :
Default TTL : 64
Arp Age : 20
IP Config [DHCP/Bootp] : Manual
IP Address : 10.35.204.104
Subnet Mask : 255.255.240.0
Online Help for Menu
Actions-> Cancel Edit Save Help
Display help information.
Use arrow keys to change action selection and <Enter> to execute action.

Figure 1-2. Online Help for Menu Interface

Command Line Interface

If you need information on a specific command in the CLI, type the command name followed by **help**. For example:

```
ProCurve# write help
Usage: write <memory|terminal>

Description: View or save the running configuration of the switch.
             write terminal - displays the running configuration of the
                              switch on the terminal
             write memory   - saves the running configuration of the
                              switch to flash. The saved configuration
                              becomes the boot-up configuration of the switch
                              the next time it is booted.
```

Figure 1-3. Example of CLI Help

Web Browser Interface

If you need information on specific features in the ProCurve Web Browser Interface, use the online Help. You can access the Help by clicking on the question mark button in the upper right corner of any of the web browser interface screens.

Figure 1-4. Button for Web Browser Interface Online Help

Note

To access the online Help for the ProCurve web browser interface, you need either ProCurve Manager (version 1.5 or greater) installed on your network or an active connection to the World Wide Web. Otherwise, Online help for the web browser interface will not be available.

To Set Up and Install the Switch in Your Network

Use the ProCurve *Installation and Getting Started Guide* (shipped with the switch) for the following:

- Notes, cautions, and warnings related to installing and using the switch and its related modules
- Instructions for physically installing the switch in your network
- Quickly assigning an IP address and subnet mask, set a Manager password, and (optionally) configure other basic features.
- Interpreting LED behavior.

For the latest version of the *Installation and Getting Started Guide* for your switch, refer to “Getting Documentation From the Web” on page 1-7.

2

Introduction to IPv6

Contents

Migrating to IPv6 2-3
- IPv6 Propagation 2-4
- Dual-Stack Operation 2-4
- Connecting to Devices Supporting IPv6 Over IPv4 Tunneling 2-5
  - Information Sources for Tunneling IPv6 Over IPv4 2-5

Use Model 2-6
- Adding IPv6 Capability 2-6
- Supported IPv6 Operation 2-6

Configuration and Management 2-7
- Management Features 2-7
- IPv6 Addressing 2-7
  - SLAAC (Stateless Automatic Address Configuration) 2-7
  - DHCPv6 (Stateful) Address Configuration 2-8
  - Static Address Configuration 2-8
  - Default IPv6 Gateway 2-8
- Neighbor Discovery (ND) in IPv6 2-9
- IPv6 Management Features 2-10
  - TFTPv6 Transfers 2-10
  - IPv6 Time Configuration 2-10
  - Telnet6 2-10
  - IP Preserve 2-11
  - Multicast Listener Discovery (MLD) 2-11
  - Web Browser Interface 2-11
- Path MTU (PMTU) Discovery 2-11

Configurable IPv6 Security 2-12
- SSHv2 on IPv6 2-12
- IP Authorized Managers 2-12

Diagnostic and Troubleshooting 2-14
ICMP Rate-Limiting 2-14
Ping6 2-14
Traceroute6 2-14
Debug/Syslog Enhancements 2-14
Domain Name System (DNS) Resolution 2-14
IPv6 Neighbor Discovery (ND) Controls 2-15
Event Log 2-15
SNMP 2-15
Loopback Address 2-15
IPv6 Scalability 2-16

Migrating to IPv6

To successfully migrate to IPv6 involves maintaining compatibility with the large installed base of IPv4 hosts and routers for the immediate future. To achieve this purpose, software release K.13.01 and greater supports dual-stack (IPv4/IPv6) operation and connections to IPv6-aware routers for routing IPv6 traffic between VLANs and across IPv4 networks.

Note

Beginning with release K.13.01 the software supports traffic connections with IPv6-aware routers, but does not support IPv6 routing operation in the switches covered by this guide.

Beginning with software release K.13.01, the switches covered by this guide support the following IPv6 protocol operations:

- receiving IPv6 traffic addressed to the switch
- transmitting IPv6 traffic originating on the switch
- switching IPv6 traffic between IPv6 devices connected to the switch on the same VLAN
- concurrent (dual-stack) operation with IPv4 traffic and devices on the same VLAN
- using a connection to an external, IPv6-configured router, forward IPv6 traffic intended for devices on other VLANs and for traffic that must traverse an IPv4 network to reach an IPv6 destination

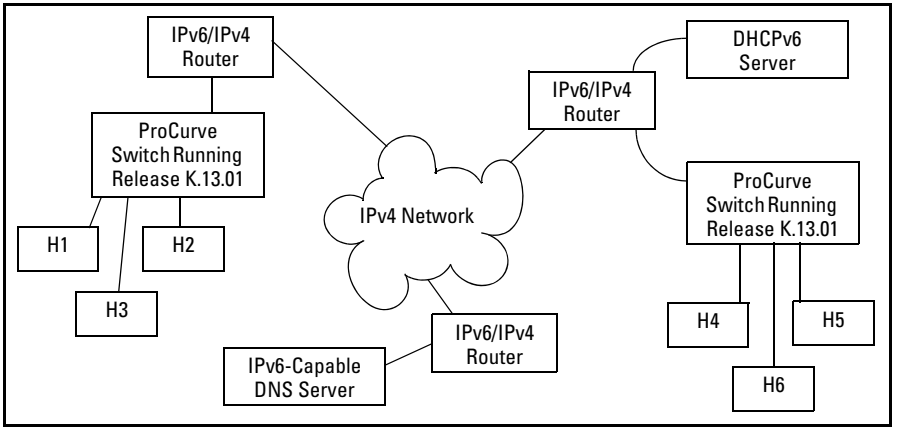

Figure 2-1. Dual-Stack ProCurve Switches Employed in an IPv4/IPv6 Network

IPv6 Propagation

IPv6 is currently in the early stages of deployment worldwide, involving a phased-in migration led by the application of basic IPv6 functionality. In these applications, IPv6 traffic is switched among IPv6-capable devices on a given LAN, and routed between LANs using IPv6-capable routers. Using the IPv6 features in this software release, the switch can operate in an IPv6 network, be managed using an IPv6 management station, and interact with DHCPv6 and IPv6-enabled DNS servers in the same network or accessible through a connection to an IPv6 router.

Dual-Stack Operation

Since most initial IPv6 deployments are in networks having a mixture of IPv6 and IPv4 hosts, software releases K.13.01 and greater support dual- stack IPv4/IPv6 operation. This enables the switch to communicate individually with IPv4 and IPv6 devices with their respective protocols. Thus, IPv4 and IPv6 traffic is supported simultaneously on the same VLAN interface. This means that both IPv4 and IPv6 devices can operate at the same time on a given VLAN.

Note

Software releases K.13.01 and greater do not include gateways for translation between IPv6 and IPv4 traffic. While IPv4 and IPv6 traffic coexists on the same VLAN, the individual IPv4 and IPv6 devices ignore each other's traffic.

To forward IPv6 traffic from the switch to an IPv6-capable device on a different VLAN, a link to an external IPv6-capable router is needed. Also, IPv6 traffic movement from the switch over IPv4 paths requires routers capable of IPv6 over IPv4 tunneling.

Connecting to Devices Supporting IPv6 Over IPv4 Tunneling

The switches covered by this guide can interoperate with IPv6/IPv4 devices capable of tunneling IPv6 traffic across an IPv4 infrastructure. Some examples include:

- traffic between IPv6/IPv4 routers (router/router)
- traffic between an IPv6/IPv4 router and an IPv6/IPv4 host capable of tunneling (router/host)

Note

Tunneling requires an IPv6-capable router. A switch running software release K.13.01 or greater does not route or tunnel IPv6 traffic. To enable IPv6 traffic from the switch to be routed or to be tunneled across an IPv4 network, it is necessary to connect the switch to an appropriate IPv6-capable router. For more information, refer to the documentation provided with the dual- stack (IPv4/IPv6) routers you plan to use for this purpose.

IPv6 tunneling eases IPv6 deployment by maintaining compatibility with the large existing base of IPv4 hosts and routers. Generally, the various IPv6 tunneling methods enable IPv6 hosts and routers to connect with other IPv6 hosts and routers over the existing IPv4 Internet.

Information Sources for Tunneling IPv6 Over IPv4

For more information on IPv6 routing and tunneling, refer to the documentation provided with the IPv6/IPv4 routing and tunneling-capable devices in your network. Some other sources of information are:

- RFC 2893: “Transition Mechanisms for IPv6 Hosts and Routers”
- RFC 2401: “Security Architecture for the Internet Protocol”
- RFC 2473: “Generic Packet Tunneling in IPv6 Specification”
- RFC 2529: “Transmission of IPv6 via IPv4 Domains without Explicit Tunnels”
- RFC 3056: “Connection of IPv6 Domains Over IPv4 Clouds”

Use Model

Adding IPv6 Capability

IPv6 was designed by the Internet Engineering Task Force (IETF) to improve on the scalability, security, ease of configuration, and network management capabilities of IPv4.

IPv6 provides increased flexibility and connectivity for existing networked devices, addresses the limited address availability inherent in IPv4, and the infrastructure for the next wave of Internet devices, such as PDAs, mobile phones and appliances.

Where IPv4 networks exist today, IPv6 will be phased in over a period of years, requiring an interoperability among the devices using the two protocols. Beginning with software release K.13.01, the switches covered by this guide support IPv4/IPv6 dual stack operation. This allows full ethernet link support for both IPv4 and IPv6 traffic to move on the same interface (VLAN) without modifying current IPv4 network topologies. This enables you to use IPv6 devices on existing VLANs, manage the switch and other devices from IPv6 management stations, and create groups of dedicated IPv6 devices as needed to accommodate the anticipated IPv6 network growth.

Supported IPv6 Operation

Software releases K.13.01 and greater provide IPv6 protocol and addressing to support host-mode (endpoint) IPv6 operation, including basic layer-2 functionality. IPv6 routing features are not available in this release. However, using a dual-stack (IPv4/IPv6-capable) router, IPv6 traffic can be routed between VLANs and sent across an IPv4 network to another IPv6 device.

(For general information on sending IPv6 traffic across an IPv4 network, refer to “Connecting to Devices Supporting IPv6 Over IPv4 Tunneling” on page 2-5.)

The next three sections outline the IPv6 features supported in software release K.13.01 and greater. These features are categorized as follows:

- configuration and management
- security
- IPv6 multicast traffic
- diagnostic and troubleshooting

Configuration and Management

This section outlines the configurable management features supporting IPv6 operation on your ProCurve IPv6-ready switch.

Management Features

Software releases K.13.01and greater provide host-based IPv6 features that enable the switches covered in this guide to be managed from an IPv6 management station and to operate in both IPv6 and IPv4/IPv6 network environments.

Note

Software releases K.13.01 and greater do not include IPv6 routing, but inter-operate with routers that support IPv6 and IPv4/IPv6 router applications.

IPv6 Addressing

The switch offers these IPv6 address configuration features:

- SLAAC (stateless automatic address configuration)
- DHCPv6 (stateful automatic address configuration)
- static address configuration

SLAAC (Stateless Automatic Address Configuration)

Enabling IPv6 on a VLAN automatically enables configuration of a link-local unicast IPv6 address on the VLAN. (No DHCPv6 server is needed.) This address begins with the hexadecimal prefix **fe80**, which is prepended to the interface identifier part of the address. (The interface identifier is generated from the MAC address of the VLAN itself, using the 64-bit extended unique identifier (EUI) method.) This enables the IPv6 nodes on the VLAN to configure and manage the switch.

Enabling IPv6 address auto configuration on a VLAN automatically enables automatic configuration of global unicast addresses on the VLAN. After enabling auto configuration, a router advertisement (RA) containing an assigned global address prefix must be received on the VLAN from an IPv6 router on the same VLAN. The resulting address is a combination of the prefix and the interface identifier currently in use in the link-local address. Having a global unicast address and a connection to an IPv6- aware router enables IPv6

traffic on a VLAN to be routed to other VLANs supporting IPv6-aware devices. (Using software release K.13.01 or greater, an external, IPv6-aware router is required to forward traffic between VLANs.)

Multiple, global unicast addresses can be configured on a VLAN that receives RAs specifying different prefixes.

DHCPv6 (Stateful) Address Configuration

The IPv6 counterpart to DHCP client for IPv4 operation is DHCPv6. Global unicast addresses of any scope can be assigned, along with NTP (timep) server addressing when DHCPv6 server support is available through either of the following modes:

- accessible on a VLAN configured on the switch
- accessible through a connection to a router configured with DHCP relay

IPv6 also allows the option of using stateless auto configuration or static configuration to assign unicast addresses to a VLAN, while using a DHCPv6 server for time server addressing.

Static Address Configuration

Statically configuring IPv6 addresses provides flexibility and control over the actual address values used on an interface. Also, if a statically configured link-local address is configured on a static VLAN, the global addresses configured on the VLAN as the result of router advertisements uses the device identifier included in the link-local address. Statically configuring an IPv6 address on a VLAN enables IPv6 on the VLAN if it has not already been enabled.

Default IPv6 Gateway

Instead of using static or DHCPv6 configuration, a default IPv6 gateway for an interface (VLAN) is determined from the default router list of reachable or probably reachable routers the switch detects from periodic multicast router advertisements (RAs) received on the interface. For a given interface, there can be multiple default gateways, with different nodes on the link using different gateways. If the switch does not detect any IPv6 routers that are reachable from a given interface, it assumes (for that interface) that it can reach only the other devices connected to the interface.

Note

In IPv6 for the switches covered in this guide, the default route cannot be statically configured. Also, DHCPv6 does not include default route configuration.)

Refer to “Default IPv6 Router” on page 4-30 and “View IPv6 Gateway, Route, and Router Neighbors ” on page 4-31.

Neighbor Discovery (ND) in IPv6

The IPv6 Neighbor Discovery protocol operates in a manner similar to the IPv4 ARP protocol to provide for discovery of IPv6 devices such as other switches, routers, management stations, and servers on the same interface. Neighbor Discovery runs automatically in the default configuration and provides services in addition to those provided in IPv4 by ARP. For example:

- Run Duplicate Address Detection (DAD) to detect duplicate unicast address assignments on an interface. An address found to be a duplicate is not used, and the **show ipv6** command displays the address as a **duplicate**.
- Quickly identify routers on an interface by sending router solicitations requesting an immediate router advertisement (RA) from reachable routers.
- If a default router becomes unreachable, locate an alternate (if available on the interface).
- Learn from reachable routers on the interface whether to use DHCPv6 or stateless address auto configuration. In the latter case, this also includes the address prefixes to use with stateless address auto configuration for routed destinations. (A DHCPv6 server can also be used for “stateless” service; that is, for configuring the interface for access to other network services, but not configuring a global IPv6 unicast address on the interface. Refer to “Neighbor Discovery (ND)” on page 4-17.)
- Use multicast neighbor solicitations to learn the link-layer addresses of destinations on the same interface and to verify that neighbors to which traffic is being sent are still reachable.
- Send a multicast neighbor advertisement in response to a solicitation from another device on the same interface or to notify neighbors of a change in the link- layer address.
- Advertise anycast addresses that may be configured on the device.
- Determine the MTU (Maximum Transmission Unit) for the interface from router advertisements.

For more on IPv6 neighbor discovery applications, refer to “Neighbor Discovery (ND)” on page 4-17.

IPv6 Management Features

The switch's IPv6 management features support operation in an environment employing IPv6 servers and management stations.With a link to a properly configured IPv6 router, switch management extends to routed traffic solutions. (Refer to the documentation provided for the IPv6 router.) Otherwise, IPv6 management for the switches covered by this guide are dependent on switched management traffic solutions.

TFTPv6 Transfers

The switch supports these downloads from an IPv6 TFTP server:

- automatic OS download
- manual OS download
- command script download and execution
- configuration file downloads
- public key file downloads
- startup configuration file downloads

The switch supports these uploads to an IPv6 TFTP server

- startup or running configuration upload
- OS upload from flash in current use (primary or secondary)
- event log content upload
- crash log content upload
- output of a specified command

Refer to “TFTP File Transfers Over IPv6” on page 5-15.

IPv6 Time Configuration

The switch supports both Timepv6 and SNTPv6 time services. Refer to “SNTP and Timep” on page 5-9.

Telnet6

The switch supports both of the following Telnet6 operations:

- Enable (the default setting) or disable Telnet6 access to the switch from remote IPv6 nodes.
- Initiate an outbound telnet session to another IPv6 networked device.

Refer to “IPv6 Telnet Operation” on page 5-6

IP Preserve

IP Preserve operation preserves both the IPv4 and IPv6 addresses configured on VLAN 1 (the default VLAN) when a configuration file is downloaded to the switch using TFTP. Refer to “IP Preserve for IPv6” on page 5-24.

Multicast Listener Discovery (MLD)

MLD operates in a manner similar to IGMP in IPv4 networks. In the factory default state (MLD disabled), the switch floods all IPv6 multicast traffic it receives on a given VLAN through all ports on that VLAN except the port receiving the inbound multicast traffic. Enabling MLD imposes management controls on IPv6 multicast traffic to reduce unnecessary bandwidth usage. MLD is configured per- VLAN. For information on MLD, refer to the chapter titled “Multicast Listener Discovery (MLD) Snooping”.

Web Browser Interface

For the web browser interface, software releases K.13.01 and greater add the following IPv6 functionality:

- configure and display IPv6 addressing
- ping6 diagnostic operation

Path MTU (PMTU) Discovery

IPv6 PMTU operation is managed automatically by the IPv6 nodes between the source and destination of a transmission. For Ethernet frames, the default MTU is 1500 bytes. If a router on the path cannot forward the default MTU size, it sends an ICMPv6 message (PKT_TOO_BIG) with the recommended MTU to the sender of the frame. If the sender of the frame is an IPv6 node that supports PMTU discovery, it will then use the MTU specified by the router and cache it for future reference.

For related information, refer to:

- RFC 1981: “Path MTU Discovery for IP version 6”

Configurable IPv6 Security

This section outlines the configurable IPv6 security features supported in software release K.14.01.

SSHv2 on IPv6

SSHv2 provides for the authentication between clients and servers, and protection of data integrity, and privacy. It is used most often to provide a secure alternative to Telnet and is also used for secure file transfers (SFTP and SCP). Beginning with software release K.13.01, SSH functionality is supported on ProCurve switches running either IPv4 or IPv6. Beginning with software release K.14.01, when SSH operation is enabled (the default setting), it automatically runs for both IPv4 and IPv6 traffic.

The switch supports up to six inbound sessions of the following types in any combination at any given time:

- SSHv2
- SSHv2 IPv6
- Telnet-server
- Telnet6-server
- SFTP/SCP (One SFTP or SCP session allowed at a given time.)
- Console (serial RS-232 connection)

For more information, refer to “Secure Shell (SSH) for IPv6” on page 6-15.

IP Authorized Managers

The IPv6 Authorized IP Managers feature, like the IPv4 version, uses IP addresses and masks to determine which stations (PCs and workstations) can access the switch through the network, and includes these access methods:

- Telnet, SSH, and other terminal emulation applications
- the switch's web browser interface
- SNMP (with a correct community name)

Also, when configured in the switch, the access control imposed by the Authorized IP Manager feature takes precedence over the other forms of access control configurable on the switch, such as local passwords, RADIUS, and both Port-Based and Client-Based Access Control (802.1X). This means

that the IP address of a networked management device must be authorized before the switch will attempt to authenticate the device by invoking any other access security features. Thus, with Authorized IP Managers configured, having the correct passwords or MAC address is not sufficient for accessing the switch through the network unless an IPv6 address configured on the station attempting the access is also included in the switch's Authorized IP Managers configuration. This presents the opportunity to combine the Authorized IP Managers feature with other access control features to enhance the security fabric protecting the switch.

Caution

The Authorized IP Managers feature does not protect against unauthorized station access through a modem or direct connection to the Console (RS-232) port. Also, if an unauthorized station "spoofs" an authorized IP address, then the unauthorized station cannot be blocked by the Authorized IP Managers feature, even if a duplicate IP address condition exists.

To configure authorized IPv6 managers, refer to "Authorized IP Managers for IPv6" on page 6-3.

For related information, refer to:

- RFC 4864, "Local Network Protection for IPv6".

Diagnostic and Troubleshooting

Software releases K.13.01 and greater include the IPv6 diagnostic and troubleshooting features listed in this section.

ICMP Rate-Limiting

Controlling the frequency of ICMPv6 error messages can help to prevent DoS (Denial- of- Service) attacks. With IPv6 enabled on the switch, you can control the allowable frequency of these messages with ICMPv6 rate-limiting. Refer to “ICMP Rate-Limiting” on page 9-2.

Ping6

Implements the Ping protocol for IPv6 destinations, and includes the same options as are available for IPv4 Ping, including DNS hostnames. Refer to “Ping for IPv6 (Ping6)” on page 9-4.

Traceroute6

Implements Traceroute for IPv6 destinations, and includes the same same options as are available for the IPv4 Traceroute, including DNS hostnames. Refer to “Traceroute for IPv6” on page 9-6.

Debug/Syslog Enhancements

Includes new options for IPv6. Refer to “Debug/Syslog for IPv6” on page 9-12.

Domain Name System (DNS) Resolution

This feature enables resolving a host name to an IPv6 address and the reverse, and takes on added importance over its IPv4 counterpart due to the extended length of IPv6 addresses. With DNS-compatible commands, CLI command entry becomes easier for reaching a device whose IPv6 address is configured with a host name counterpart on a DNS server.

Software release K.13.01 includes the following DNS-compatible commands:

- **ping6**
- **traceroute6**

The switches covered by this guide now support a prioritized list of up to three DNS server addresses. (Earlier software releases supported only one DNS server address.) Also, the server address list can include both IPv4 and IPv6 DNS server addresses. (An IPv6 DNS server can respond to IPv4 queries, and the reverse.)

Note

If an IPv6 DNS server address is configured on the switch, at least one VLAN on the switch (and in the path to the DNS server) must be configured with an IPv6 address.

For information on configuring DNS resolution on the switch, refer to "DNS Resolver for IPv6" on page 9-9.

IPv6 Neighbor Discovery (ND) Controls

The neighbor discovery feature includes commands for:

- increasing or decreasing the frequency of Duplicate Address Detection searches
- displaying the IPv6 neighbor cache
- clearing dynamic entries from the neighbor cache

Refer to "Neighbor Discovery (ND) in IPv6" on page 2-9.

Event Log

Messages returning IP addresses now include IPv6 addresses where applicable.

SNMP

When IPv6 is enabled on a VLAN interface, you can manage the switch from a network management station configured with an IPv6 address. Refer to "SNMP Management for IPv6" on page 5-21.

Loopback Address

Like the IPv4 loopback address, the IPv6 loopback address (::1) can be used by the switch to send an IPv6 packet to itself. However, the IPv6 loopback address is implicit on a VLAN and cannot be statically configured on any VLAN. Refer to "Loopback Address" on page 3-24.

IPv6 Scalability

As of software release K.14.01, the switches covered by this guide support the following:

- Dual stack operation (IPv4 and IPv6 addresses on the same VLAN).
- per-switch

VLANs, maximum configured	2048
VLANs, maximum with IPv4 and IPv6 addresses in any combination	512
IP addresses	IPv4: 2048 IPv6 user-configured: 2048 IPv6 auto-configured: 2048*
IP addresses per-VLAN	IPv4: 32 IPv6 user-configured: 32 IPv6 auto-configured, prefix based: 3
IPv6 routes	10,000

*Auto-configured link-local and prefix-based addresses.

- Maximum of 2048 active IPv6 addresses on the switch, in addition to a maximum of 2048 IPv4 addresses. (“Active IPv6 addresses” includes the total of all preferred and non-preferred addresses configured statically, through DHCPv6, and through stateless auto configuration. Excluded from “Active IPv6 Addresses” is the link-local address assigned to each VLAN, and “on- link” prefixes received as part of a router advertisement.)
- Maximum of 10,000 IPv6 routes.

For more information on VLAN and route scalability on the switches covered by this guide, refer to the appendix titled “Scalability: IP Address, VLAN, and Routing Maximum Values” in the *Management and Configuration Guide* for your switch.

IPv6 Addressing

Contents

Introduction 3-3

IPv6 Address Structure and Format 3-3

Address Format 3-3

Address Notation 3-3

Network Prefix 3-4

Interface (Device) Identifier 3-4

IPv6 Addressing Options 3-5

IPv6 Address Sources 3-5

General IPv6 Address Types 3-5

IPv6 Address Sources 3-7

Stateless Address Autoconfiguration (SLAAC) 3-7

Applications 3-7

Preferred and Valid Lifetimes of Stateless Autoconfigured Addresses 3-7

Stateful (DHCPv6) Address Configuration 3-8

Static Address Configuration 3-9

Address Types and Scope 3-10

Address Types 3-10

Address Scope 3-11

Unicast Address Prefixes 3-11

Link-Local Unicast Address 3-13

Autoconfiguring Link-Local Unicast Addresses 3-13

Extended Unique Identifier (EUI) 3-14

Statically Configuring Link-Local Addresses 3-15

Global Unicast Address 3-16

Stateless Autoconfiguration of a Global Unicast Address 3-16

Static Configuration of a Global Unicast Address 3-17

Prefixes in Routable IPv6 Addresses 3-18
Unique Local Unicast IPv6 Address 3-19
Anycast Addresses 3-20
Multicast Application to IPv6 Addressing 3-21
Overview of the Multicast Operation in IPv6 3-21
IPv6 Multicast Address Format 3-22
Multicast Group Identification 3-22
Solicited-Node Multicast Address Format 3-23
Loopback Address 3-24
The Unspecified Address 3-25
IPv6 Address Deprecation 3-25
Preferred and Valid Address Lifetimes 3-25

Introduction

IPv6 supports multiple addresses on an interface, and uses them in a manner comparable to subnetting an IPv4 VLAN. For example, where the switch is configured with multiple VLANs and each is connected to an IPv6 router, each VLAN will have a single link-local address and one or more global unicast addresses. This section describes IPv6 addressing and outlines the options for configuring IPv6 addressing on the switch. The configuration process includes automatically or statically creating an IPv6 address and automatically verifying the uniqueness of each.

IPv6 Address Structure and Format

Address Format

An IPv6 address is composed of 128 bits divided into eight 2-byte fields of hexadecimal values. The full format is:

XXXX : XXXX : XXXX : XXXX : XXXX : XXXX : XXXX : XXXX

where each field delimited by a colon (:) is a set of four hexadecimal digits.

For example:

2001:0db8:0000:00A9:0215:60ff:fe7a:adc0

2001:0db8:0260:0212:0000:0000:0000:01b4

The hexadecimal characters in IPv6 addresses are not case-sensitive.

Address Notation

Leading zeros in each field can be omitted as long as each field is represented by at least one value. The exception to this rule is when there is an uninterrupted series of zeros in one or more contiguous fields. In this case, the series of zeros can be replaced by "::", with the restriction that "::" can be used only once in a given address. Applying this convention to the above examples results in the following address notations:

2001:db8::a9:215:60ff:fe7a:adc0

2001:db8:260:0212::01b4

An IPv6 address includes a network prefix and an interface identifier.

Network Prefix

The network prefix (high-order bits) in an IPv6 address begins with a well-known, fixed prefix for defining the address type. Some examples of well-known, fixed prefixes are:

2000::/3global (routable) unicast address

fd08::/8 unique local unicast address

fe80::/8 link-local unicast address

ff00::/8 multicast address

The remainder of the network prefix depends on the prefix type, and includes information such as the subnet destination of unicast addresses or the flags and scope of multicast addresses.

In a given address, CIDR-type notation (Classless Inter-Domain Routing) is used to define the network prefix. In the following address example, the 64 bits comprising 2001:0db8:0260:0212 form the network prefix:

2001:0db8:0260:0212:0215:60ff:fe7a:adc0/64

A shorter way to show this address is to remove the leading zeros:

2001:db8:260:212:215:60ff:fe7a:adc0/64

Interface (Device) Identifier

The remaining (low-order) bits in the address comprise a unique interface identifier in an IPv6 address. In the above example, the rightmost 64 bits (215:60ff:fe7a:adc0) comprise the interface identifier. Unlike IPv4, an IPv6 identifier for a unicast or anycast address can be automatically generated from the switch's MAC address using EUI-64 (Extended Unique Identifier) format. Other methods include DHCPv6 assignments and static configuration. Interface identifiers are covered in more detail in the later sections of this chapter describing different address types.

IPv6 Addressing Options

IPv6 Address Sources

IPv6 addressing sources provide a flexible methodology for assigning addresses to VLAN interfaces on the switch. Options include:

- stateless IPv6 Autoconfiguration on VLAN interfaces includes:
  - link-local unicast addresses
  - global unicast addresses
- stateful, global unicast IPv6 address configuration using DHCPv6
- static IPv6 address configuration

You can combine stateless, stateful, and static IP addressing methods on the switch as needed, according to the needs in your network. For example, if your network includes only one VLAN, you may need only stateless Autoconfiguration of link-local addresses, although you could also use the static IPv6 method. (DHCPv6 does not configure link-local addresses.) Where routed traffic is used, you will also need global unicast addressing, either through stateless Autoconfiguration or the other listed methods.

General IPv6 Address Types

IPv6 supports stateless and stateful address Autoconfiguration, as well as static address configuration.This enables IPv6 to automatically address a device so that it can be placed in a network with or without static or DHCPv6 addressing intervention. All three of these methods can be used exclusively or in conjunction with each other, and a given IPv6 device can have multiple addresses assigned to the same interface in a manner similar to subnetting in IPv4.

Stateless Address Autoconfiguration. This method does not require the use of servers. Instead, in the default operation, the host uses its MAC address to automatically generate a link-local IPv6 address using the EUI-64 method to generate the device identifier. (Refer to “Autoconfiguring Link-Local Unicast Addresses” on page 3-13.) The scope of the link-local address enables communication with other IPv6 devices on the same VLAN. If an IPv6 router is present, an IPv6 address supporting routing is automatically generated, as well. (The switch merges a router-generated prefix received in router advertisements with the last 64 bits of the link-local address on an interface to create the global address.) Refer to page 3-7.

Stateful Address Autoconfiguration. This method allows use of a DHCPv6 server to automatically configure IPv6 addressing on a host in a manner similar to stateful IP addressing with a DHCPv4 server. For software releases K.13.01 and greater, a DHCPv6 server can provide routable IPv6 addressing and NTP (timep) server addresses. Also, if the host acquires its IPv6 addressing through stateless or static methods, the DHCPv6 server can still be used to automatically provide other configuration information to the host. Refer to page 3-8.

Static Address Configuration. Static configuration is used instead of or in addition to stateless and stateful Autoconfiguration where use of the host MAC address does not provide the desired level of address control and distribution. Refer to page 3-9.

Duplicate Address Detection (DAD). IPv6 verifies both the link-local and the global unicast address(es) on each interface for uniqueness, regardless of the method used to configure the address. If an address fails this test, it is identified as a **duplicate**, and a replacement must be configured using the static method. (To view address status, use the **show ipv6** command.) For more information on DAD, refer to "Neighbor Discovery (ND)" on page 4-17.

Developing an Addressing Plan. For small, flat networks and any environment where control of address assignments need not be restricted or tightly controlled, stateless addressing is adequate for network management and control. Where systematic and controlled addressing is needed, stateful and static addressing methods should be used. Where dual-stack operation is used in a VLAN, incorporating the local IPv4 addressing scheme into the IPv6 addresses you use can help to provide consistency and correspondence among the IPv6 and IPv4 addresses in use on the VLAN.

Related Information.

- RFC 4291: "IP Version 6 Addressing Architecture"
- RFC 2462: "IPv6 Stateless Address Autoconfiguration"
- RFC 3315: "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)"

IPv6 Address Sources

IPv6 addressing sources provide a flexible methodology for assigning addresses to VLAN interfaces on the switch. Options include:

- stateless IPv6 Autoconfiguration on VLAN interfaces includes:
  - link-local unicast addresses
  - global unicast addresses
- stateful IPv6 address configuration using DHCPv6
- static IPv6 address configuration

You can combine stateless, stateful, and static IP addressing methods on the switch as needed, according to the needs in your network. For example, if your network includes only one VLAN, you may need only stateless Autoconfiguration of link-local addresses, although you could also use the static IPv6 method. (DHCPv6 does not configure link-local addresses.) Where routed traffic is used, you will also need global unicast addressing, either through stateless Autoconfiguration or the other listed methods.

Stateless Address Autoconfiguration (SLAAC)

On the switches covered by this guide, stateless address Autoconfiguration (SLAAC) generates link-local unicast and global unicast IPv6 addresses on a VLAN interface. In all cases, the prefix is 64 bits.

Applications

Stateless Autoconfiguration is suitable where a link-local or global unicast IPv6 address (if a router is present) must be unique, but the actual address used is not significant. Where a specific unicast address or a unicast address from a specific range of choices is needed on an interface, DHCPv6 or static IPv6 address configuration should be used. (Refer to pages 3-8 and 3-9.)

Preferred and Valid Lifetimes of Stateless Autoconfigured Addresses

The preferred and valid lifetimes of an Autoconfigured global unicast address are set by the router advertisements (RA) used to generate the address, and are the Autoconfiguration counterpart to the lease time assigned by DHCPv6

servers. These lifetimes cannot be reset using control from the switch console or SNMP methods. Refer to "Preferred and Valid Address Lifetimes" on page 3-25.

Stateful (DHCPv6) Address Configuration

Stateful addresses are defined by a system administrator or other authority, and automatically assigned to the switch and other devices through the Dynamic Host Configuration Protocol (DHCPv6). Generally, DHCPv6 should be applied when you want specific, non-default addressing to be assigned automatically. For IPv6, DHCP use is indicated for conditions such as the following:

- address conventions used in your network require defined control
- static addressing is not feasible due to the number of nodes in the network
- automatic assignment of multiple IPv6 addresses per interfaces is needed
- automatic configuration of IPv6 access to DNS, SNTP, or TimeP servers

To implement stateful address configuration:

- The DHCPv6 server must be configured and accessible to the switch, either on the same VLAN or through an IPv6 router configured with DHCP Relay to support service requests from the switch.

Note

DHCPv6 relay may not currently be available in some IPv6 routers.

- DHCPv6 addressing must be enabled per-VLAN on the switch.

Note that IPv6 router advertisements (RAs) can also include instructions to clients to use DHCPv6 resources. Refer to the documentation for your IPv6 router.

If you want to use DHCPv6 in a dual-stack environment, you will need both DHCPv4 and DHCPv6 server access. Also, further developments in DHCP services are likely to mean new capabilities affecting DHCPv6 deployments.

For related information, refer to:

- RFC 3315: "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)"
- RFC 3041: "Privacy Extensions for Stateless Address Autoconfiguration in IPv6"

Static Address Configuration

Generally, static address configuration should be used when you want specific, non-default addressing to be assigned to a VLAN interface. For IPv6, DHCP use is indicated for conditions such as the following:

- address conventions used in your network require defined control
- the task of static addressing is not so extensive as to be impractical due to the number of addresses and/or interfaces needing configuration

If IPv6 is not already enabled on a VLAN interface, the following is true:

- Statically configuring a link-local address on the interface also enables IPv6.
- Statically configuring a global unicast or anycast address also enables IPv6 and generates a link-local address.

Statically configured global unicast addresses can be used in addition to stateless addresses on the same interface. However, because only one link-local address is allowed on a VLAN interface (fe80::), static configuration of a link-local address automatically replaces an existing link-local address.

Note

For a statically configured global unicast address to be routable, a gateway router must be transmitting router advertisements on the VLAN that include the prefix used in the statically configured address. If the VLAN is not receiving an RA with this prefix, the address is listed as “preferred”, but is not used.

Statically configured IPv6 addresses saved to the startup-config file (by using **write memory**) remain across a reboot and are permanent, unless statically removed by **no ipv6 address < *ipv6-addr* >**.

For more information and the CLI command for static address configuration, refer to “Configuring a Static IPv6 Address on a VLAN” on page 4-11.

Address Types and Scope

Address Types

IPv6 uses these IP address types:

- **Unicast:** Identifies a specific IPv6 interface. Traffic having a unicast destination address is intended for a single interface. Like IPv4 addresses, unicast addresses can be assigned to a specific VLAN on the switch and to other IPv6 devices connected to the switch. At a minimum, a given interface must have at least a link-local address. To send or receive traffic off of a VLAN, an interface must also have one or more global unicast addresses.
- **Multicast:** Provides a single destination address for traffic intended for all members of a group, and provides a means for reducing unnecessary traffic to interfaces that do not belong to a given multicast group. Membership in a group can be determined by request or by a characteristic, such as all nodes, all routers, or all routers of a given type. Multicast traffic can be generated by a single source or multiple sources, but in either case is intended for multiple destinations.Common types of multicast traffic include streaming video and audio to multiple receivers who have joined a specific group from diverse locations.

Note

Unlike IPv4, broadcast addresses are not used in IPv6. Multicast addresses are used instead. For more on this topic, refer to “Multicast Application to IPv6 Addressing” on page 3-21.

- **Anycast:** A single address of this type can be assigned to multiple interfaces, possibly on separate devices within a defined address scope, where any of the interfaces having the anycast address can provide the desired service or response. A packet sent to a given anycast address is delivered only to the nearest interface having an instance of the address. This option is useful where multiple servers provide the same service, and it does not matter to the client which source it uses to acquire the service. Anycast usage can be of value, for example, in a network supporting multiple DNS servers. Refer to “Anycast Addresses” on page 3-20.

A given interface can have only one link-local address, but can have multiple unicast and anycast addresses.

Address Scope

The address scope determines the area (topology) in which a given IPv6 address is used. This section provides an overview of IPv6 address types. For more information, refer to the chapter titled “IPv6 Addressing”.

Link-Local Address. Limited to a given interface (VLAN). Enabling IPv6 on a given VLAN automatically generates a link-local address used for switched traffic on the VLAN.

Global Unicast Address. Applies to a unique IPv6 routable address on the internet. A unique global address has a routing prefix and a unique device identifier.When Autoconfiguration is enabled on a VLAN receiving an IPv6 router advertisement (RA), the prefix specified in the RA and the device identifier specified in the link-local address are combined to create a unique, global unicast address. A global unicast address can also be statically configured to either replace or complement an automatically configured address of the same type.

Unique Local Unicast. Applies to a routable, globally unique address intended for use within an entity defined by the system administrator, such as a specific site or a group of related sites defined by IPv6 border routers. These addresses are intended to be routable on a local site or an organization's intranet, but are not intended to be routed on the global internet. A unique local unicast address has the same format as a global unicast address. In this guide, unless otherwise stated, information on global unicast addresses also applies to unique local unicast addresses. For more on this topic, refer to “Unique Local Unicast IPv6 Address” on page 3-19.

Unicast Address Prefixes

Traffic having a unicast destination address is intended for a single interface identified by that address. While IPv6 unicast addresses can have prefixes of varying length, a 64-bit prefix is generally adequate.

Link-Local Unicast Prefix (fe80): This well-known 64-bit fixed prefix is for a non- routable address used to identify a device on a single VLAN interface, and requires the high-order ten bits to be set to fe80 (fe80::/10). The remaining 54 bits in the prefix are set to zeros, followed by an interface ID of 64 bits.

fe80:0000:0000:0000:0215:60ff:fe7a:adc0/64

or

fe80::215:60ff:fe7a:asc0/64

In binary notation, the fixed prefix for link-local prefixes is:

1111 1110 10 = fe80/10

For more on link-local addresses, refer to “Link-Local Unicast Address” on page 3-13.

Routable Global Unicast Prefix. This well-known 3-bit fixed-prefix indicates a routable address used to identify a device on a VLAN interface that is accessible by routing from multiple networks. The complete prefix is 64 bits, followed by a 64-bit interface identifier. For example, the leading 2 in the first octet of the following address illustrates a global unicast address:

2001:db8:260:212:215:60ff:fe7a:adc0/64

In binary notation, the fixed prefix in this example appears as follows:

0010 0000 = 20/3

Unique Local Unicast Prefix (fd). This well-known fixed prefix is defined as FC00/7. However, the eighth high-order bit must also be set to 1, resulting in a fixed prefix of fd00/8. (In the future, setting the eighth high-order bit to zero may become an option.) This prefix signifies a routable address intended for use within the boundaries of a site or organization. For example, the leading fd in the first octet of this address illustrates a unique local unicast address intended to be used in a privately defined network.

fd00:00ff:0C00:000a:215:60ff:fe7a:adc0

Unique local unicast addresses are described in more detail under "Unique Local Unicast IPv6 Address" on page 3-19.

Multicast Prefix (ff). This well-known 8-bit fixed prefix signifies a permanent or temporary multicast address. The second 8 high-order bits are used for flags and scope for the multicast address. The remaining 112 bits define the multicast group identifier. For example:

ff02::1:ffc7:b5b9

For more information, refer to “Multicast Application to IPv6 Addressing” on page 3-21.

Other Prefix Types. There are other designated global unicast prefixes such as those for the following address types:

- RFC 4380: “Teredo: Tunneling IPv6 over UDP”
- RFC 3056: “Connection of IPv6 Domains via IPv4 Clouds”
- RFC 4214: “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)”

For related information, refer also to:

- RFC 4291: "IP Version 6 Addressing Architecture

Link-Local Unicast Address

A link-local unicast address is a non-routable address for use on a single VLAN interface, and provides basic connectivity to an IPv6 network. Because the scope of a link-local address is restricted to the VLAN on which the address is used, a link-local address must be unique only for the VLAN on which it is configured. (Traffic with a link-local source or destination address cannot be routed between VLANs.)

Autoconfiguring Link-Local Unicast Addresses

Enabling IPv6 on a given VLAN automatically generates a link-local address. This address is limited in scope to that VLAN, and is usable only for switched traffic. This address has a well- known, 64-bit prefix of fe80:0000:0000:0000 (hexadecimal), or fe80::, and a 64-bit device identifier derived from the VLAN's MAC address using the Extended Unique Identifier format (EUI-64, page 3-14). For example, if the MAC address of VLAN 10 is 021560-7aadc0, the automatically generated link-local address for VLAN 10 is:

fe80:0000:0000:0000:0215:60ff:fe7a:adc0

or, in standard IPv6 notation,

fe80::215:60ff:fe7a:adc0

Note that only one link-local address is allowed on an interface. Thus, on a given interface, statically configuring a link-local address type replaces the existing link-local address.

Because all VLANs configured on the switch use the same MAC address, all automatically generated link-local addresses on the switch will have the same link-local address. However, since the scope of a link-local address includes only the VLAN on which it was generated, this should not be a problem.

For example, executing **ipv6 address dhcp full** on a VLAN for which IPv6 was not previously configured does all of the following:

- enables IPv6 on the VLAN
- causes the switch to generate a stateless link-local unicast address on the VLAN
- configures the VLAN to send DHCPv6 requests

Note

Only one link-local unicast address can exist on a VLAN interface at any time. Configuring a new address of this type on an interface on which IPv6 is already enabled replaces the previously existing link-local address with the new one.

Any link-local address must include the well-known link-local prefix fe80::/64 plus a 64-bit device identifier.

Any of the following commands enable IPv6 on a VLAN and automatically generate a link-local address:

- **ipv6 enable** (page 4-6)
- **ipv6 address autoconfig** (page 4-7)
- **ipv6 address dhcp full [rapid-commit]** (page 4-9)
- **ipv6 address < *network-prefix*><*device-id* >/< *prefix-length* >** (page 4-13)

Extended Unique Identifier (EUI)

When the link-local address is automatically generated, the device identifier is derived from the switch's 48- bit (hexadecimal) MAC address to create a 64-bit Extended Unique Identifier (EUI) to be appended to the fe80 link-local prefix, as follows:

- ff-fe is inserted between third and fourth bytes of MAC address
- The second low-order bit (the Universal/Local bit) in the first byte of the MAC address is complemented, which usually means the bit is originally set to 0 and is changed to 1. This indicates a globally unique IPv6 interface identifier. For example:

MAC Address	IPv6 I/F Identifier	Full Link-Local Unicast Address
00-15-60-7a-ad-c0	215:60ff:fe7a:adc0	fe80::215:60ff:fe7a:adc0/64
09-c1-8a-44-b4-9d	11c1:8aff:fe44:b49d	fe80::11c1:8aff:fe44:b49d/64
00-1a-73-5a-7e-57	21a:73ff:fe5a:7e57	fe80::21a:73ff:fe5a:7e57/64

The EUI method of generating a link-local address is automatically implemented on the switches covered by this guide when IPv6 is enabled on a VLAN interface.

If automatically generated link-local addresses are not suitable for the addressing scheme you want to use, statically assigned link-local addresses can be used instead. (Refer to “Static Address Configuration” on page 3-9.)

For related information, refer to:

- RFC 2373: “IP Version 6 Addressing Architecture”
- RFC 2464: “Transmission of IPv6 Packets Over Ethernet Networks”

Note

While only one link-local IPv6 address is allowed on an interface, multiples of other address types can exist on the same interface. Thus, an interface can have one link-local unicast address, but multiple global unicast, anycast, and unique local addresses.

Statically Configuring Link-Local Addresses

A link-local unicast address can be configured statically on a VLAN interface. If IPv6 is not already enabled on the VLAN, this action also enables IPv6 on the VLAN. Only one link-local address can exist on a VLAN at any time. If a link-local address (static or Autoconfigured) already exists on the VLAN, then statically configuring a new one replaces the previously existing one. To statically configure a link-local address, refer to “Statically Configuring a Link-Local Unicast Address ” on page 4-12.

Global Unicast Address

A global unicast address is required for unicast traffic to be routed across VLANs within an organization as well as across the public internet. To support subnetting, a VLAN can be configured with multiple global unicast addresses. Any of the following methods can be used to configure this kind of address on a VLAN:

- stateless address Autoconfiguration using a prefix received in an advertisement received from a router on the VLAN (page 3-7)
- stateful address configuration using DHCPv6 (page 3-8)
- static address configuration (page 3-9)

Stateless Autoconfiguration of a Global Unicast Address

If there is an IPv6-enabled router transmitting router advertisements on a VLAN interface, enabling this method generates a global, routable unicast address for the VLAN. The prefix for this address type is typically 64 bits with the three highest-order bits set to 2.

Router Advertisements. With Autoconfiguration enabled, if the switch receives the same prefix from router advertisements (RAs) from multiple IPv6 routers on the same VLAN, then one global unicast address is configured with that prefix. If different prefixes are received from different routers on the same VLAN, then there will be one address configured on the VLAN for each unique prefix received. Where there are multiple routers on the VLAN, the default route for the VLAN is determined by the relative router priorities included in the RAs the VLAN receives. If the highest priority is duplicated on multiple routers, then the first RA detected on the VLAN determines the default route.

If the RA used to define the prefix for an Autoconfigured address ceases to be received on the VLAN, then the address becomes deprecated. (Refer to “IPv6 Address Deprecation” on page 3-25.)

If IPv6 is not already enabled on a VLAN when you enable Autoconfiguration on the VLAN, then the switch automatically generates a link-local address for the VLAN as well.

If IPv6 Is Not Already Enabled. Enabling address Autoconfiguration on a VLAN when IPv6 is not already enabled on the VLAN causes the switch to:

- generate a link-local address on the VLAN as described in the preceding section (page 3-13).
- transmit a router solicitation on the VLAN, and to listen for advertisements from any IPv6 routers on the VLAN.

For each unique router advertisement (RA) the switch receives from any router(s), the switch configures a unique, global unicast address. This address type is composed of a 64-bit network prefix specified by the router advertisement, plus a device identifier generated in the same way as described in the proceeding section for link-local addresses (using the EUI algorithm). For example, suppose the following is true:

- IPv6 is not enabled on VLAN 1.
- The MAC address for VLAN 1 is 00-15-60-7a-ad-c0.
- A router on the same VLAN transmits router advertisements that assign the prefix 2001:0:260:212/64, plus a 64-bit interface identifier generated using the EUI format.

In this case, enabling IPv6 address Autoconfiguration on VLAN 1 generates the following address assignments on VLAN 1:

- link-local unicast: fe80::215:60ff:fe7a:adc0/64
- global unicast:2001:0:260:212:215:60ff:fe7a:adc0/64

IPv6 Already Enabled. Enabling address Autoconfiguration on a VLAN when IPv6 is already enabled on the VLAN creates a global unicast address in the same way as described above, except that the device identifier applied to the new global address is a duplicate of the 64-bit identifier in the current link-local address.

Note

After a global unicast address has been configured, its device identifier will not be changed by any later changes to the link-local address.

Static Configuration of a Global Unicast Address

A global unicast address can be configured statically on a VLAN interface. If IPv6 is not already enabled on a VLAN, then statically configuring a global unicast address automatically generates a link-local unicast address on the VLAN, as described in the preceding section. To statically configure a global unicast address, refer to “Statically Configuring A Global Unicast Address” on page 4-13.

Prefixes in Routable IPv6 Addresses

In routable IPv6 addresses, the prefix uniquely identifies an entity and a unicast subnet within that entity, and is defined by a length value specifying the number of leftmost contiguous (high-order) bits comprising the prefix. For an automatically generated global unicast address, the default prefix length is 64 bits. (Practically speaking, the entire prefix in a /64 address defines the subnet.) Prefixes configured through stateful or static methods can be any length compatible with the local network application.

In the following example, the leftmost 64 bits of the address comprise the prefix:

2001:0db8:0000:0212:0215:60ff:fe7a:adc0/64

or

2001:db8::212:215:60ff:fe7a:adc0/64

In this case, the prefix is read as:

2001:0db8:0000:0212::

or

2001:db8::212::

All bits to the right of 0212 comprise the device identifier in the unicast address.

For related information, refer to:

- RFC 3177: “IAB/IESG Recommendations on IPv6 Address Allocations to Sites”
- RFC 4291: “IP Version 6 Addressing Architecture”

Unique Local Unicast IPv6 Address

A unique local unicast address is an address that falls within a specific range, but is used only as a global unicast address within an organization. Traffic having a source address within the defined range should not be allowed beyond the borders of the intended domain or onto the public internet.

The current prefix for specifically identifying unique local unicast addresses is fd00/8. The leftmost 64 bits of a unique local unicast address include:

- the well-known prefix “fd”
- a 40-bit global identifier
- a 16-bit subnet identifier

For example:

fd73:110:255:23:215:60ff:fe7a:adc0/64

In the above case, the following values are used with the well-known prefix and L-bit setting:

- global identifier: 0073:110:255
- subnet identifier: 23
- interface identifier: 215:60ff:fe7a:adc0

Unique local unicast addresses can be assigned by router advertisements, DHCPv6 servers, or static configuration. The boundaries for unique local unicast address are set by border routers. Unique local unicast addresses can be assigned in DNS servers supporting an internal network, but should not be included in global DNS assignments.

For related information, refer to:

- RFC 4193: “Unique Local IPv6 Unicast Addresses”

Anycast Addresses

Network size, traffic loads and the potential for network changes make it desirable to build in redundancy for some network services to provide increased service reliability. Anycast addressing provides this capability for applications where it does not matter which source is actually used to provide a service that is offered on multiple sources. Some applications that can benefit from anycast addressing include:

- DNS (UDP)
- time servers
- multicast rendezvous
- syslog devices
- gateways to a common network area.

Similarly, it is also useful in some cases to economically provide redundant paths to a given entity, such as a specific service provider. With IPv6 this can be done efficiently using the anycast address capability to assign the same address to multiple devices providing access to the desired services. An added benefit of utilizing anycast addresses is to reduce the need to configure clients with the addresses of multiple devices offering the same service.

An anycast address is an identifier for a set of interfaces typically belonging to different nodes. Packets sent to an anycast address are delivered to one of the interfaces identified as the “nearest” address, according to the routing protocol's measure of distance.

Note Equal-Cost paths between a host and multiple instances of the same anycast address can result in different packets in the same communication session to be sent to different destinations, and should be avoided.

An anycast address is formatted the same as a unicast address. For this reason, configuring an anycast address on the switch includes using an **anycast** keyword as part of the command. The prefix for an anycast address should include all areas of the network in which the address is used. For information on configuring an anycast address on the switches covered by this guide, refer to “Statically Configuring An Anycast Address” on page 4-14.

Note Duplicate Address Detection (DAD) does not apply to anycast addresses.

For related information, refer to:

- RFC 4291: "IP Version 6 Addressing Architecture"
- RFC 2526: "Reserved IPv6 Subnet Anycast Addresses"

Multicast Application to IPv6 Addressing

Multicast is used to reduce traffic for applications that have more than one recipient for the same data. IPv6 also uses multicast for purposes such as providing a more defined control of administrative traffic on a VLAN interface than can be achieved with the broadcast method used by IPv4. This approach improves traffic control for such purposes as neighbor and router solicitations, router advertisements, and responses to DAD messages. It also avoids the bandwidth consumption used for broadcasts by narrowing the scope of possibly interested destinations for various types of messages.

Overview of the Multicast Operation in IPv6

When IPv6 is enabled on a VLAN interface on the switch, the interface automatically joins the *All-Nodes* and *Solicited-Node* multicast address groups for each of its configured unicast and anycast addresses. The interface also attempts to learn of other devices by sending solicitations to additional, well-known multicast groups, such as the following:

- all routers
- all MLDv2-capable routers, if multicast listener discovery (MLD) is enabled on the interface
- all DHCP agents (if DHCP is enabled on the interface)

There is a separate, *solicited node multicast group* for each IPv6 unicast and anycast address configured on a given interface. These automatically generated groups are limited in scope to the VLANs on which the node resides. Where multiple IPv6 unicast or anycast addresses on the same node differ only in their prefixes, they join the same solicited-node multicast group. Solicited-Node multicast groups are used, for example, in Autoconfiguration. In this case, a node attempting to Autoconfigure a link-local address computes the solicited-node multicast address for the proposed link-local address, then sends a Neighbor solicitation to this solicited-node multicast address. If there is no response from another node, the proposed address is available for use.

For more on Neighbor Discovery, refer to "Neighbor Discovery (ND)" on page 4-17.

For information on Multicast Listener Discovery (MLD) refer to the chapter titled “Multicast Listener Discovery (MLD) Snooping”.

When MLD is enabled on an interface, you can use **show ipv6 mld [vlan < *vid* >]** to list the active multicast group activity the switch has detected per interface from other devices.

IPv6 Multicast Address Format

The multicast address format has three principal sections in the leading 16 bits:

- identifier: ff (bits 1-8)
- flags: 0xxx (bits 9-12)
- scope: 0001 - 1110 (bits 13-16)

For related information, refer to RFC 4291.

Multicast Group Identification

Multicast ID, Flags and Scope (16 bits)	Group Identifier (112 bits)
1111 1111 0*xxx xxxx*:	*x...x* : *x...x* : *x...x* : *x...x* : *x...x* : *x...x* : *x...x* : *x...x*

- **multicast identifier:** The first eight high-order bits, set to ff, identify the address as multicast.
- **multicast flags:** Bits 9-12 are multicast flags that provide additional information about the multicast address, as follows:

Bit ID	Options	Use
9	0	reserved
10 (R)	0	multicast address without PIM-SM rendezvous point
	1	multicast address with PIM-SM rendezvous point
11 (P)	0	multicast address without prefix information from the originating network
	1	multicast address with prefix information from the originating network
12 (T)	0	multicast address is permanent (well-known, and not restricted by scope value)
	1	multicast address is temporary (and used only within an identified scope)

- **multicast scope:** Bits 13-16 set boundaries on multicast traffic distribution, such as the interface defined by the link-local unicast address of an area, or the network boundaries of an organization. Because IPv6 uses multicast technology in place of the broadcast technology used in IPv4, the multicast scope field also controls the boundaries for broadcast-type traffic sent in multicast packets.

Bit	Use
0	reserved
1	interface-local (loopback)
2	link-local (same topology as the corresponding link-local unicast scope)
3	reserved
4	admin-local (smallest administratively configured scope)
5	site-local (single site)
6	*unassigned*
7	*unassigned*
8	organization-local (multiple sites within the same organization)
9	*unassigned*
A	*unassigned*
B	*unassigned*
C	*unassigned*
D	*unassigned*
E	global
F	reserved

  For example, the following prefix indicates multicast traffic with a temporary multicast address and a link-local scope:

  ff12 or (binary) 1111 1111 0001 0010

- **group identifier:** This field includes the last 112 bits of the multicast address and contains the actual multicast group identity. (Refer to RFCs 3306, 4291, and 2375.)

Solicited-Node Multicast Address Format

The solicited-node multicast address the switch generates for a configured unicast or anycast address is composed of a unique, 104-bit multicast prefix (ff02:0:0:0:0:1:ff) and the last 24 bits of the subject address. For example, if a VLAN interface is configured with a link-local address of

fe90::215:60ff:fe7a:adc0

then the corresponding solicited-node multicast address is

ff02:0:0:0:0:1:ff7a:adc0

For related information, refer to:

- RFC 2375: IPv6 Multicast Address Assignments
- RFC 3306: Unicast-Prefix-based IPv6 Multicast Addresses
- RFC 3956: Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address
- RFC 3177: IAB/IESG Recommendations on IPv6 Address Allocations to Sites
- RFC 4007: IPv6 Scoped Address Architecture
- RFC 4291: IP Version 6 Addressing Architecture
- "Internet Protocol Version 6 Multicast Addresses" (at **www.iana.org**)
- RFC 2710: Multicast Listener Discovery (MLD) for IPv6
- RFC 3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6 (Updates RFC 2710.)

Loopback Address

The IPv6 loopback address is a link-local unicast address that enables a device to send traffic to itself for self-testing purposes. The loopback address does not have a physical interface assignment. If an IPv6 packet destined for the loopback address is received on a switch interface, it must be dropped. The IPv6 loopback address is never used as the source IPv6 address for any packet that is sent out of a device, and the switch drops any traffic it receives with a loopback address destination. An example use case is:

```
ProCurve# ping6 ::1

0000:0000:0000:0000:0000:0000:0000:0001 is alive, time = 1 ms
```

The Unspecified Address

The “unspecified” address is defined as 0.0.0.0.0.0.0.0 (::/128, or just ::). It can be used, for example, as a temporary source address in multicast traffic sent by an interface that has not yet acquired its own address. The unspecified address cannot be statically configured on the switch, or used as a destination address.

IPv6 Address Deprecation

Preferred and Valid Address Lifetimes

Autoconfigured IPv6 global unicast addresses acquire their valid and preferred lifetime assignments from router advertisements. A *valid* lifetime is the time period during which an address is allowed to remain available and usable on an interface. A *preferred* lifetime is the length of time an address is intended for full use on an interface, and must be less than or equal to the address's valid lifetime.

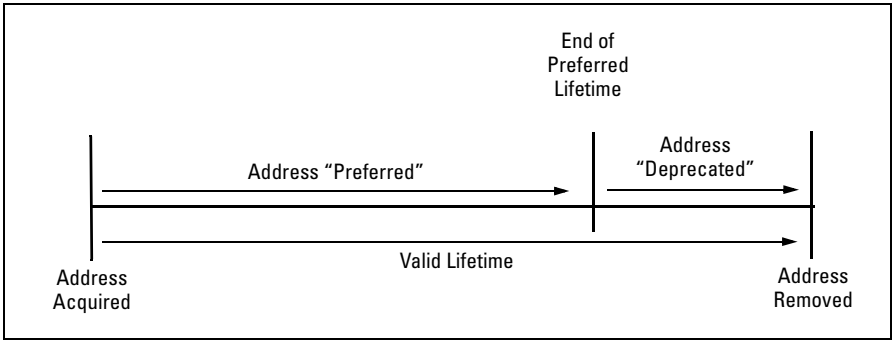

Figure 3-1. Valid and Preferred Lifetimes

When the preferred lifetime expires, the address becomes *deprecated*, meaning that the address should no longer be used as a source address (except for existing exchanges that began before the timeout occurred), but can still be used as a destination. When the timeout arrives for the valid lifetime, the address becomes unusable.

Notes

Preferred and valid lifetimes on a VLAN interface are determined by the router advertisements received on the interface. These values are not affected by the lease time assigned to an address by a DHCPv6 server. That is, lease expiration on a DHCPv6-assigned address terminates use of the address, regardless of the status of the RA-assigned lifetime, and router-assigned lifetime expiration of a leased address terminates the switch's use of the address. (The router-assigned lifetime can be extended by receipt of a new router advertisement.)

Statically configured IPv6 addresses are regarded as permanent addresses, and do not expire.

Related Information

- RFC 2462: "IPv6 Stateless Address Autoconfiguration"
- RFC 4291: "IP Version 6 Addressing Architecture"

IPv6 Addressing Configuration

Contents

Introduction 4-3

General Configuration Steps 4-4

Configuring IPv6 Addressing 4-5

Enabling IPv6 with an Automatically Configured Link-Local Address 4-6

Enabling Autoconfiguration of a Global Unicast Address and a Default Router Identity on a VLAN 4-7

Operating Notes 4-8

Enabling DHCPv6 4-9

Operating Notes 4-10

Configuring a Static IPv6 Address on a VLAN 4-11

Statically Configuring a Link-Local Unicast Address 4-12

Statically Configuring A Global Unicast Address 4-13

Operating Notes 4-14

Statically Configuring An Anycast Address 4-14

Duplicate Address Detection (DAD) for Statically Configured Addresses 4-16

Disabling IPv6 on a VLAN 4-16

Neighbor Discovery (ND) 4-17

Duplicate Address Detection (DAD) 4-18

DAD Operation 4-18

Configuring DAD 4-19

Operating Notes for Neighbor Discovery 4-20

View the Current IPv6 Addressing Configuration 4-22

Router Access and Default Router Selection 4-29

Router Advertisements 4-29

Router Solicitations 4-29
Default IPv6 Router 4-30
Router Redirection 4-30
View IPv6 Gateway, Route, and Router Neighbors 4-31
Viewing Gateway and IPv6 Route Information 4-31
Viewing IPv6 Router Information 4-32
Address Lifetimes 4-34
Preferred Lifetime 4-34
Valid Lifetime 4-34
Sources of IPv6 Address Lifetimes 4-34

Introduction

Feature	Default	CLI
Enable IPv6 with a Link-Local Address	disabled	4-6
Configure Global Unicast Autoconfig	disabled	4-7
Configure DHCPv6 Addressing	disabled	4-9
Configure a Static Link-Local Address	None	4-12
Configure a Static Global Unicast Address	None	4-13
Configure an Anycast Address	None	4-14
Change DAD Attempts	3	4-18
View Current IPv6 Addressing	*n/a*	4-22

In the default configuration, IPv6 operation is disabled on the switch. This section describes the general steps and individual commands for enabling IPv6 operation.

This chapter provides the following:

- general steps for IPv6 configuration
- IPv6 command syntax descriptions, including **show** commands

Most IPv6 configuration commands are applied per-VLAN. The exceptions are ICMP, ND (neighbor discovery), and the (optional) authorized-managers feature, which are configured at the global configuration level. (ICMP and ND for IPv6 are enabled with default values when IPv6 is first enabled, and can either be left in their default settings or reconfigured, as needed.) For more information on ICMP, refer to "ICMP Rate-Limiting" on page 9-2. For more on ND, refer to "Neighbor Discovery (ND) in IPv6" on page 2-9.

For a quick reference to all IPv6 commands available on the switch, refer to the "IPv6 Command Index" on page xv at the front of this guide.

Note

Beginning with software release K.13.01, the switch is capable of operating in dual-stack mode, where IPv4 and IPv6 run concurrently on a given VLAN.

General Configuration Steps

The IPv6 configuration on switches running software release K.13.01 or greater includes global and per-VLAN settings. This section provides an overview of the general configuration steps for enabling IPv6 on a given VLAN and can be enabled by any one of several commands. The following steps provide a suggested progression for getting started.

Note The ICMP and Neighbor Discovery (ND) parameters are set to default values at the global configuration level are satisfactory for many applications and generally do not need adjustment when you are first configuring IPv6 on the switch.

In the default configuration, IPv6 is disabled on all VLANs.

1. If IPv6 DHCP service is available, enable IPv6 DHCP on the VLAN. If IPv6 is not already enabled on the VLAN, enabling DHCPv6 also enables IPv6 and automatically configures a link-local address using the EUI-64 format.

Note If IPv6 is not already enabled on the VLAN, enabling DHCPv6 causes the switch to automatically generate a link-local address. DHCPv6 does not assign a link-local address.

   A DHCPv6 server can provide other services, such as the addresses of time servers. For this reason you may want to enable DHCP even if you are using another method to configure IPv6 addressing on the VLAN.

2. If IPv6 DHCP service is not enabled on the VLAN, then do either of the following:
   - Enable IPv6 on the VLAN. This automatically configures a link-local address with an EUI- 64 interface identifier.
   - Statically configure a unicast IPv6 address on the VLAN. This enables IPv6 on the VLAN and, if you configure anything other than a link-local address, the link-local address will be automatically configured as well, with an EUI-64 interface identifier.
3. If an IPv6 router is connected on the VLAN, then enable IPv6 address autoconfiguration to automatically configure global unicast addresses with prefixes included in advertisements received from the router. The device identifier used in addresses configured by this method will be the same as the device identifier in the current link-local address.

4. If needed, statically configure IPv6 unicast addressing on the VLAN interface as needed. This can include any of the following:
   - statically replacing the automatically generated link-local address
   - statically adding global unicast, unique local unicast, and/or anycast addresses

Configuring IPv6 Addressing

In the default configuration on a VLAN, any one of the following commands enables IPv6 and creates a link-local address. Thus, while any one of these methods is configured on a VLAN, IPv6 remains enabled and a link-local address is present:

ipv6 enable (page 4-6)

ipv6 address autoconfig (page 4-7)

ipv6 address dhcp full [rapid-commit] (page 4-9)

ipv6 address fe80:0:0:0:< *device-identifier* > link-local (page 4-12)

ipv6 address < *prefix:device-identifier* > (page 4-13)

Note

Addresses created by any of these methods remain tentative until verified as unique by Duplicate Address Detection. (Refer to “Duplicate Address Detection (DAD)” on page 4-18.)

Enabling IPv6 with an Automatically Configured Link-Local Address

This command enables automatic configuration of a link-local address .

Syntax: [no] ipv6 enable

If IPv6 has not already been enabled on a VLAN by another IPv6 command option described in this chapter, this command enables IPv6 on the VLAN and automatically configures the VLAN's link-local unicast address with a 64-bit EUI-64 interface identifier generated from the VLAN MAC address. (Refer to "Extended Unique Identifier (EUI)" on page 3-14.).

Note: *Only one link-local IPv6 address is allowed on the VLAN interface. Subsequent static or DHCP configuration of another link-local address overwrites the existing link-local address.*

A link-local address always uses the prefix fe80:0:0:0.

With IPv6 enabled, the VLAN uses received router advertisements to designate the default IPv6 router. (Refer to "Default IPv6 Router" on page 4-30.)

After verification of uniqueness by DAD, a link-local IPv6 address assigned automatically is set to the **preferred** *status, with a "permanent" lifetime. (Refer to "IPv6 Address Deprecation" on page 3-25.)*

Default: *Disabled*

The **no** *form of the command disables IPv6 on the VLAN if no other IPv6-enabling command is configured on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 4-16.)*

To view the current IPv6 Enable setting and any statically configured IPv6 addresses per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 4-22.

Enabling Autoconfiguration of a Global Unicast Address and a Default Router Identity on a VLAN

Enabling autoconfig or rebooting the switch with autoconfig enabled on a VLAN causes the switch to configure IPv6 addressing on the VLAN using router advertisements and an EUI-64 interface identifier (page 3-14).

Syntax: [no] ipv6 address autoconfig

Implements unicast address autoconfiguration as follows:

- *If IPv6 is not already enabled on the VLAN, this command enables IPv6 and generates a link-local (EUI- 64) address.*
- *Generates router solicitations (RS) on the VLAN.*
- *If a router advertisement (RA) is received on the VLAN, the switch uses the route prefix in the RA to configure a global unicast address. The device identifier for this address will be the same as the device identifier used in the current link-local address at the time the RA is received. (This can be either a statically configured or the (automatic) EUI-64 device identifier, depending on how the link-local address was configured.) For information on EUI- 64, refer to "Extended Unique Identifier (EUI)" on page 3-14.) If an RA is not received on the VLAN after autoconfig is enabled, a link-local address will be present, but no global unicast addresses will be autoconfigured.*

Notes: *If a link-local address is already configured on the VLAN, a later, autoconfigured global unicast address uses the same device identifier as the link-local address.*

Autoconfigured and DHCPv6-assigned global unicast addresses with the same prefix are mutually exclusive on a VLAN. On a given switch, if both options are configured on the same VLAN, then only the first to acquire a global unicast address will be used.

— Continued on the next page. —

— Continued from the previous page. —

After verification of uniqueness by DAD, an IPv6 address assigned to a VLAN by autoconfiguration is set to the preferred and valid lifetimes specified by the RA used to generate the address, and is configured as a preferred address. (Refer to "IPv6 Address Deprecation" on page 3-25.)

Default: *Disabled.*

The **no** *form of the command produces different results, depending on how IPv6 is configured on the VLAN:*

If IPv6 was enabled only by the **autoconfig** *command, then deleting this command disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 4-16.)*

To view the current IPv6 autoconfiguration settings per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 4-22.

Operating Notes

With IPv6 enabled, the VLAN uses received router advertisements to designate the default IPv6 router. (Refer to "Router Access and Default Router Selection" on page 4-29.)

Enabling DHCPv6

Enabling the DHCPv6 option on a VLAN allows the switch to obtain a global unicast address and an NTP (network time protocol) server assignment for a Timep server. (If a DHCPv6 server is not needed to provide a global unicast address to a switch interface, the server can still be configured to provide the NTP server assignment. This is sometimes referred to as "stateless DHCPv6".)

Syntax: [no] ipv6 address dhcp full [rapid-commit]

This option configures DHCPv6 on a VLAN, which initiates transmission of DHCPv6 requests for service. If IPv6 is not already enabled on the VLAN by the **ipv6 enable** *command, this option also enables IPv6 and causes the switch to autoconfigure a link-local unicast address with an EUI-64 interface identifier.*

> ***Notes:*** *A DHCPv6 server does not assign link-local addresses, and enabling DHCPv6 on a VLAN does not affect a pre-existing link-local address configured on the VLAN.*
>
> *A DHCPv6-assigned address can be configured on a VLAN when the following is true:*
>
> - *The assigned address is not on the same subnet as a previously configured autoconfig address.*
> - *The maximum IPv6 address limit on the VLAN or the switch has not been reached.*

If a DHCPv6 server responds with an IPv6 address assignment, this address is assigned to the VLAN. (The DHCPv6-assigned address will be dropped if it has the same subnet as another address already assigned to the VLAN by an earlier autoconfig command.)

— Continued on the next page. —

— Continued from the previous page. —

After verification of uniqueness by DAD, an IPv6 address assigned to the VLAN by an DHCPv6 server is set to the preferred and valid lifetimes specified in a router advertisement received on the VLAN for the prefix used in the assigned address, and is configured as a preferred address. (Refer to the section titled "Address Lifetimes" on page 4-34.)

[rapid-commit]: *Expedites DHCP configuration by using a two-message exchange with the server (solicit-reply) instead of the default four-message exchange (solicit-advertise- request-reply).*

Default: *Disabled*

The **no** *form of the command removes the DHCPv6 option from the configuration and, if no other IPv6-enabling command is configured on the VLAN, disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 4-16.)*

To view the current IPv6 DHCPv6 settings per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on the VLAN.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 4-22.

Operating Notes

- If multiple DHCPv6 servers are available, the switch selects a server based on the preference value sent in DHCPv6 messages from the servers.
- The switch supports both DHCPv4 and DHCPv6 client operation on the same VLAN.
- DHCPv6 authentication and stateless DHCPv6 are not supported in software releases K.13.01 or greater.
- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the router advertisements it receives. (Refer to "Default IPv6 Router" on page 4-30.)

- DHCPv6 and statically configured global unicast or anycast addresses are mutually exclusive on a given VLAN. That is, configuring DHCPv6 on a VLAN erases any static global unicast or anycast addresses previously configured on that VLAN, and the reverse. (A statically configured link-local address will not be affected by configuring DHCPv6 on the VLAN.)
- For the same subnet on the switch, a DHCPv6 global unicast address assignment takes precedence over an autoconfigured address assignment, regardless of which address type was the first to be configured. If DHCPv6 is subsequently removed from the configuration, then an autoconfigured address assignment will replace it after the next router advertisement is received on the VLAN. DHCPv6 and autoconfigured addresses co-exist on the same VLAN if they belong to different subnets.

For related information refer to:

- RFC 3315: “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”
- RFC 3633: “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6”
- RFC 3736: “Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6”

Configuring a Static IPv6 Address on a VLAN

This option enables configuring of unique, static unicast and anycast IPv6 addresses for global and link-local applications, including:

- link-local unicast (including EUI and non-EUI device identifiers)
- global unicast (and unique local unicast)
- anycast

Statically Configuring a Link-Local Unicast Address

Syntax: [no] ipv6 address fe80::< *device-identifier* > link-local

- *If IPv6 is not already enabled on the VLAN, this command enables IPv6 and configures a static link-local address.*
- *If IPv6 is already enabled on the VLAN, then this command overwrites the current, link- local address with the specified static address. (One link-local address is allowed per VLAN interface.)*

< *device-identifier* >: The low-order 64 bits, in 16-bit blocks, comprise this value in a link-local address:

xxxx xxxx : xxxx xxxx : xxxx xxxx : xxxx xxxx

Where a static link-local address is already configured, a new, autoconfigured global unicast addresses assignment uses the same device identifier as the link-local address.

> ***Notes:*** *An existing link-local address is replaced, and is not deprecated, when a static replacement is configured.*
>
> *The prefix for a statically configured link-local address is always 64 bits, with all blocks after fe80 set to zero. That is: fe80:0:0:0.*

After verification of uniqueness by DAD, a statically configured link-local address status is set to **preferred**, *with a* **permanent** *lifetime. (Refer to "IPv6 Address Deprecation" on page 3-25.)*

For link-local addressing, the ***no*** *form of the static IPv6 address command produces different results, depending on how IPv6 is configured on the VLAN:*

- *If IPv6 was enabled only by a statically configured link-local address, then deleting the link-local address disables IPv6 on the VLAN.*
- *If other IPv6-enabling commands have been configured on the VLAN, then deleting the statically configured link-local address causes the switch to replace it with the default (EUI-64) link-local address for the VLAN, and IPv6 remains enabled. (For more on the EUI-64 address format, refer to "Extended Unique Identifier (EUI)" on page 3-14.)*

Refer also to "Disabling IPv6 on a VLAN" on page 4-16.

Statically Configuring A Global Unicast Address

Syntax:. [no] ipv6 address < network-prefix><device-id >/< prefix-length >
[no] ipv6 address < network-prefix>::/< prefix-length > eui-64

If IPv6 is not already enabled on a VLAN, either of these command options do the following:

- *enable IPv6 on the VLAN*
- *configure a link-local address using the EUI-64 format*
- *statically configure a global unicast address*

If IPv6 is already enabled on the VLAN, then the above commands statically configure a global unicast address, but have no effect on the current link-local address.

< *network-prefix* >: *This includes the global routing prefix and the subnet ID for the address. For more on this topic, refer to "Prefixes in Routable IPv6 Addresses" on page 3-18.*

< *device-id* >: Enters a user-defined device identity.

< *prefix-length* >: Specifies the number of bits in the network prefix. If you are using the **eui-64** option, this value must be 64.

eui-64: *Specifies using the Extended Unique Identifier format to create a device identifier based on the VLAN MAC address. Refer to "Extended Unique Identifier (EUI)" on page 3-14.*

After verification of uniqueness by DAD, the lifetime of a statically configured IPv6 address assigned to a VLAN is set to permanent, and is configured as a preferred address. (Refer to "IPv6 Address Deprecation" on page 3-25.)

The **no** *form of the command erases the specified address and, if no other IPv6-enabling command is configured on the VLAN, disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 4-16.)*

To view the currently configured static IPv6 addresses per-VLAN, use **show run**.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on **VLAN < *vid* >**.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 4-22.

Operating Notes

- With IPv6 enabled, the switch determines the default IPv6 router for the VLAN from the router advertisements it receives. (Refer to "Router Access and Default Router Selection" on page 4-29.)
- If DHCPv6 is configured on a VLAN, then configuring a static global unicast address on the VLAN removes DHCPv6 from the VLAN's configuration and deletes the DHCPv6-assigned global unicast address.
- Note that for a statically configured global unicast address to be routable, a gateway router must be transmitting router advertisements on the VLAN.
- If an autoconfigured global unicast address already exists for the same subnet as a new, statically configured global unicast address, the statically configured address is denied. In the reverse case, you can add an autoconfig command to the VLAN configuration, but it will not be implemented unless the static address is removed from the configuration.

Statically Configuring An Anycast Address

Anycast addresses on the switch appear the same as global unicast addresses. To configure an anycast address on a VLAN, append the **anycast** keyword to the same command that is used to statically configure a global unicast address. (Link-Local unicast addresses cannot be configured as anycast addresses on the switch.)

Anycast addresses are allocated from the unicast address space, and cannot be distinguished from other IPv6 global unicast addresses configured on the switch, except by viewing the address configurations listed per-VLAN in the **show run** output. For more information on using anycast addresses, refer to "Anycast Addresses" on page 3-20.

Syntax:. [no] ipv6 address < *network-prefix* >< *device-identifier* >/< *prefix-length* > anycast

If IPv6 is not already enabled on a VLAN, this command option does the following:

- *enables IPv6 on the VLAN*
- *configures a link-local address using the EUI-64 format*
- *statically configures an anycast address*

If IPv6 is already enabled on the VLAN, then the above commands statically configure an anycast address, but has no effect on the current link-local address.

anycast: *Identifies the specified address as an anycast address. This allows the address to be duplicated (as an anycast address) on other devices on the same network.*

Default: *None.*

The **no** *form of the command erases the specified anycast address and, if no other IPv6- enabling command is configured on the VLAN, disables IPv6 on the VLAN. (Refer to "Disabling IPv6 on a VLAN" on page 4-16.)*

To verify the identity of anycast addresses configured for VLANs to which the switch belongs, use the **show run** command.

To view all currently configured IPv6 unicast addresses, use the following:

- **show ipv6** (Lists IPv6 addresses for all VLANs configured on the switch.)
- **show ipv6 vlan < *vid* >** (Lists IPv6 addresses configured on **VLAN < *vid* >**.)

For more information, refer to "View the Current IPv6 Addressing Configuration" on page 4-22.

Duplicate Address Detection (DAD) for Statically Configured Addresses

Statically configured IPv6 addresses are designated as permanent. If DAD determines that a statically configured address duplicates a previously configured and reachable address on another device belonging to the VLAN, then the more recent, duplicate address is designated as **duplicate**. For more on this topic, refer to:

- "Duplicate Address Detection (DAD)" on page 4-18.
- "View the Current IPv6 Addressing Configuration" on page 4-22

Note

Multiple, duplicate addresses configured as Anycast on different devices are special cases of unicast addresses, and are not identified as duplicates by DAD. Refer to "Anycast Addresses" on page 3-20.

Disabling IPv6 on a VLAN

While one IPv6-enabling command is configured on a VLAN, IPv6 remains enabled on that VLAN. In this case, removing the only IPv6-enabling command from the configuration disables IPv6 operation on the VLAN. That is, to disable IPv6 on a VLAN, all of the following commands must be removed from the VLAN's configuration:

```
ipv6 enable
ipv6 address dhcp full [rapid-commit]
ipv6 address autoconfig
ipv6 address fe80::< device-identifier > link-local
ipv6 address < prefix > : < device-identifier >
```

If any of the above remain enabled, then IPv6 remains enabled on the VLAN and, at a minimum, a link-local unicast address will be present.

Neighbor Discovery (ND)

Neighbor Discovery (ND) is the IPv6 equivalent of the IPv4 ARP for layer 2 address resolution, and uses IPv6 ICMP messages to do the following:

- Determine the link-layer address of neighbors on the same VLAN interface.
- Verify that a neighbor is reachable.
- Track neighbor (local) routers.

Neighbor Discovery enables functions such as the following:

- router and neighbor solicitation and discovery
- detecting address changes for devices on a VLAN
- identifying a replacement for a router or router path that has become unavailable
- duplicate address detection (DAD)
- router advertisement processing
- neighbor reachability
- autoconfiguration of unicast addresses
- resolution of destination addresses
- changes to link-layer addresses
- anycast address operation

An instance of Neighbor Discovery is triggered on a device when a new (tentative) or changed IPv6 address is detected. (This includes stateless, stateful, and static address configuration.) ND operates in a per-VLAN scope; that is, within the VLAN on which the the device running the ND instance is a member. Neighbor discovery actually occurs when there is communication between devices on a VLAN. That is, a device needing to determine the link-layer address of another device on the VLAN initiates a (multicast) neighbor solicitation message (containing a solicited-node multicast address that corresponds to the IPv6 address of the destination device) on the VLAN. When the destination device receives the neighbor solicitation, it responds with a neighbor advertisement message identifying its link-layer address. When the initiating device receives this advertisement, the two devices are ready to exchange traffic on the VLAN interface. Also, when an IPv6 interface becomes operational, it transmits a router solicitation on the interface and listens for a router advertisement.

Note: Neighbor and router solicitations must originate on the same VLAN as the receiving device. To support this operation, IPv6 is designed to discard any incoming neighbor or router solicitation that does not have a value of 255 in the IP Hop Limit field. For a complete list of requirements, refer to RFC 246.

When a pair of IPv6 devices in a VLAN exchange communication, they enter each other's IPv6 and corresponding MAC addresses in their respective neighbor caches. These entries are maintained for a period of time after communication ceases, and then dropped.

To view or clear the content of the neighbor cache, refer to “Viewing and Clearing the IPv6 Neighbors Cache” on page 5-2.

For related information, refer to:

- RFC 2461: “Neighbor Discovery for IP Version 6 (IPv6)”

Duplicate Address Detection (DAD)

Duplicate Address Detection verifies that a configured unicast IPv6 address is unique before it is assigned to a VLAN interface on the switch. DAD is enabled in the default IPv6 configuration, and can be reconfigured, disabled, or re-enabled at the global config command level. DAD can be useful in helping to troubleshoot erroneous replies to DAD requests, or where the neighbor cache contains a large number of invalid entries due to an unauthorized station sending false replies to the switch's neighbor discovery queries. If DAD verifies that a unicast IPv6 address is a duplicate, the address is not used. If the link-local address of the VLAN interface is found to be a duplicate of an address for another device on the interface, then the interface stops processing IPv6 traffic.

DAD Operation

On a given VLAN interface, when a new unicast address is configured, the switch runs DAD for this address by sending a neighbor solicitation to the All-Nodes multicast address (ff02::1). This operation discovers other devices on the VLAN and verifies whether the proposed unicast address assignment is unique on the VLAN. (During this time, the address being checked for uniqueness is held in a tentative state, and cannot be used to receive traffic other than neighbor solicitations and neighbor advertisements.) A device that receives the neighbor solicitation responds with a Neighbor Advertisement

that includes its link-local address. If the newly configured address is from a static or DHCPv6 source and is found to be a duplicate, it is labelled as duplicate in the "Address Status" field of the **show ipv6** command, and is not used. If an autoconfigured address is found to be a duplicate, it is dropped and the following message appears in the Event Log:

```
W < date > < time > 00019 ip: ip address < IPv6-address >
removed from vlan id < vid >
```

DAD does not perform periodic checks of existing addresses. However, when a VLAN comes up with IPv6 unicast addresses configured (as can occur during a reboot) the switch runs DAD for each address on the interface by sending neighbor solicitations to the All-Nodes multicast address as described above.

If an address is configured while DAD is disabled, the address is assumed to be unique and is assigned to the interface. If you want to verify the uniqueness of an address configured while DAD was disabled, re-enable DAD and then either delete and reconfigure the address, or reboot the switch.

Configuring DAD

Syntax: ipv6 nd dad-attempts < 0 - 600 >

This command is executed at the global config level, and configures the number of neighbor solicitations to send when performing duplicate address detection for a unicast address configured on a VLAN interface.

< 0 - 600 >: *The number of consecutive neighbor solicitation messages sent for DAD inquiries on an interface. Setting this value to 0 disables DAD on the interface. Disabling DAD bypasses checks for uniqueness on newly configured addresses. If a reboot is performed while DAD is disabled, the duplicate address check is not performed on any IPv6 addresses configured on the switch.*

Default: *3 (enabled); Range: 0 - 600 (0 = disabled)*

The **no** form of the command restores the default setting (3).

Syntax: ipv6 nd ns-interval < *milliseconds* >

Used on VLAN interfaces to reconfigure the neighbor discovery time in milliseconds between DAD neighbor solicitations sent for an unresolved destination, or between duplicate address detection neighbor solicitation requests. Increasing this setting is indicated where neighbor solicitation retries or failures are occurring, or in a "slow" (WAN) network .

To view the current setting, use **show ipv6 nd**.

Range: *1000 - 3600000 ms;* ***Default:*** *1000 ms.*

Syntax: ipv6 nd reachable-time < *milliseconds* >

Used on VLAN interfaces to configure the length of time in milliseconds a neighbor will be considered reachable after the Neighbor Unreachability Detection algorithm has confirmed it to be reachable. When the switch operates in host mode, this setting can be overridden by a reachable time received in a router advertisement.

To view the current setting, use **show ipv6 nd**.

Range: *1000 - 2147483647 ms;* ***Default:*** *30000 ms.*

Operating Notes for Neighbor Discovery

- A verified link-local unicast address must exist on a VLAN interface before the switch can run DAD on other addresses associated with the interface.
- If a previously configured unicast address is changed, a neighbor advertisement (an all-nodes multicast message–ff02::1) is sent to notify other devices on the VLAN and to perform duplicate address detection.
- IPv6 addresses on a VLAN interface are assigned to multicast address groups identified with well- known prefixes. For more on this topic, refer to "Multicast Application to IPv6 Addressing" on page 3-21.
- DAD is performed on all stateful, stateless, and statically configured unicast addresses, but not on Anycast addresses.
- Neighbor solicitations for DAD do not cause the neighbor cache of neighboring switches to be updated.

- If a previously configured unicast address is changed, a neighbor advertisement is sent on the VLAN to notify other devices, and also for duplicate address detection.
- If DAD is disabled when an address is configured, the address is assumed to be unique and is assigned to the interface.

View the Current IPv6 Addressing Configuration

Use these commands to view the current status of the IPv6 configuration on the switch.

Syntax: show ipv6

Lists the current, global IPv6 settings and per-VLAN IPv6 addressing on the switch.

IPv6 Routing: *For software releases K.13.01 through K.14.01, this setting is always* **Disabled**. *This is a global setting, and is not configured per-VLAN. (Refer to "Router Access and Default Router Selection" on page 4-29.)*

Default Gateway: *Lists the IPv4 default gateway, if any, configured on the switch. This is a globally configured router gateway address, and is not configured per-VLAN.*

ND DAD: *Indicates whether DAD is enabled (the default) or disabled. Using* **ipv6 nd dad-attempts 0** *disables neighbor discovery. (Refer to "Duplicate Address Detection (DAD)" on page 4-18.)*

DAD Attempts: *Indicates the number of neighbor solicitations the switch transmits per-address for duplicate (IPv6) address detection. Implemented when a new address is configured or when an interface with configured addresses comes up (such as after a reboot). The default setting is 3, and the range is 0 - 600. A setting of "0" disables duplicate address detection. (Refer to "Duplicate Address Detection (DAD)" on page 4-18.)*

VLAN Name: *Lists the name of a VLAN statically configured on the switch.*

IPv6 Status: *For the indicated VLAN, indicates whether IPv6 is disabled (the default) or enabled. (Refer to "Configuring IPv6 Addressing" on page 4-5.)*

Address Origin:

- **Autoconfig:** *The address was configured using stateless address autoconfiguration (SLAAC). In this case, the device identifier for global unicast addresses copied from the current link-local unicast address.*
- **DHCP:** *The address was assigned by a DHCPv6 server. Note that addresses having a DHCP origin are listed with a 128-bit prefix length.*
- **Manual:** *The address was statically configured on the VLAN.*
- **IPv6 Address/Prefix Length:** *Lists each IPv6 address and prefix length configured on the indicated VLAN.*

Address Status:

- **Tentative:** *DAD has not yet confirmed the address as unique, and is not usable for sending and receiving traffic.*
- **Preferred:** *The address has been confirmed as unique by DAD, and usable for sending and receiving traffic. The Expiry time shown for this address by the* **show ipv6 vlan < *vid* >** *command output is the preferred lifetime assigned to the address. (Refer to "Address Lifetimes" on page xxx.)*
- **Deprecated:** *The preferred lifetime for the address has been exceeded, but there is time remaining in the valid lifetime.*
- **Duplicate:** *Indicates a statically configured IPv6 address that is a duplicate of another IPv6 address that already exists on another device belonging to the same VLAN interface. A duplicate address is not used.*

For example, figure 4-1 shows the output on a switch having IPv6 enabled on one VLAN.

```
ProCurve# show ipv6

 Internet (IPv6) Service

  IPv6 Routing    : Disabled
  Default Gateway : fe80::213:c4ff:fedd:14b0
  ND DAD          : Enabled
  DAD Attempts    : 3

  Vlan Name       : DEFAULT_VLAN
  IPv6 Status     : Disabled

  Vlan Name       : VLAN10
  IPv6 Status     : Enabled

  Address    |                                          Address
  Origin     | IPv6 Address/Prefix Length               Status
  ---------- + ---------------------------------------- -----------
  dhcp       | 2001:db8:a03:e102::1:101/64              preferred
  manual     | fe80::1:101/64                           preferred
```

Figure 4-1. Example of Show IPv6 Command Output

Syntax: show ipv6 nd

Displays the current IPv6 neighbor discovery settings on the configured VLAN interfaces.

For example, figure 4-25 shows the output on a switch having IPv6 enabled on VLANs 1 and 20.

```
ProCurve# show ipv6 nd

 IPV6 Neighbor Discovery Configuration


  Current Hop Limit  : 0

  VLAN Name    RCHtime  NSint
               (msecs)  (msecs)
  ------------ -------- --------
  DEFAULT_VLAN 30000    1000
  VLAN20       30000    1000
```

Figure 4-2. Example of Show IPv6 nd Output with Default settings

Syntax: show ipv6 vlan < *vid* >

Displays IP and IPv6 global configuration settings, the IPv6 status for the specified VLAN, the IPv6 addresses (with prefix lengths) configured on the specified VLAN, and the expiration data (Expiry) for each address.:

- **IPv6 Routing:** *For software releases K.13.01 through K.14.01, this setting is always* **Disabled**. *(Refer to "Router Access and Default Router Selection" on page 4-29.).*
- **Default Gateway:** *Lists the IPv4 default gateway, if any, configured on the switch. This is a globally configured router gateway address, and is not configured per-VLAN.*
- **ND DAD:** *Shows whether Neighbor Discovery (ND) is enabled. The default setting is Enabled. Using* **ipv6 nd dad-attempts 0** *disables neighbor discovery.*

- **DAD Attempts:** *Indicates the number of neighbor solicitations the switch transmits per-address for duplicate (IPv6) address detection. Implemented when a new address is configured or when an interface with configured addresses comes up (such as after a reboot). The default setting is 3, and the range is 0 - 600. A setting of "0" disables duplicate address detection. (Refer to "Duplicate Address Detection (DAD)" on page 4-18.)*
- **VLAN Name:** *Lists the name of a VLAN statically configured on the switch.*
- **IPv6 Status:** *For the indicated VLAN, indicates whether IPv6 is disabled (the default) or enabled. (Refer to "Configuring IPv6 Addressing" on page 4-5.)*
- **IPv6 Address/Prefix Length:** *Lists each IPv6 address and prefix length configured on the indicated VLAN.*
- **Expiry:** *Lists the lifetime status of each IPv6 address listed for a VLAN:*
  - **Permanent:** *The address will not time out and need renewal or replacement.*
  - **date/time:** *The date and time that the address expires. Expiration date and time is specified in the router advertisement used to create the prefix for automatically configured, global unicast addresses. The* **Address Status** *field in the* **show ipv6** *command output indicates whether this date/time is for the "preferred" or "valid" lifetime assigned to the corresponding address. (Refer to "Preferred and Valid Address Lifetimes" on page 3-25.)*

```
ProCurve# show ipv6 vlan 10

 Internet (IPv6) Service

  IPv6 Routing    : Disabled
  Default Gateway : fe80::213:c4ff:fedd:14b0%vlan10
  ND DAD          : Enabled
  DAD Attempts    : 3

  Vlan Name       : VLAN10
  IPv6 Status     : Enabled


  IPv6 Address/Prefixlength                   Expiry
  ------------------------------------------- -------------------------
  2001:db8:a03:e102::1:101/64                 Fri May 19 11:51:15 2009

  fe80::1:101/64                              permanent
```

Figure 4-3. Example of Show IPv6 VLAN < *vid* > Output

Syntax: show run

In addition to the other elements of the current configuration, this command lists the statically configured, global unicast and anycast IPv6 addressing, and the current IPv6 configuration per-VLAN. The listing may include one or more of the following, depending on what other IPv6 options are configured on the VLAN. Any stateless address autoconfiguration (SLAAC) commands in the configuration are also listed in the output, but the actual addresses resulting from these commands are not included in the output.

- ipv6 enable
- ipv6 address fe80::< *device-id* > link-local
- ipv6 address < *prefix* >:< *device-id* >/< *prefix-length* >
- ipv6 address autoconfig
- ipv6 address dhcp full [rapid-commit]
- ipv6 < *global-unicast-address* >/< *prefix* > anycast

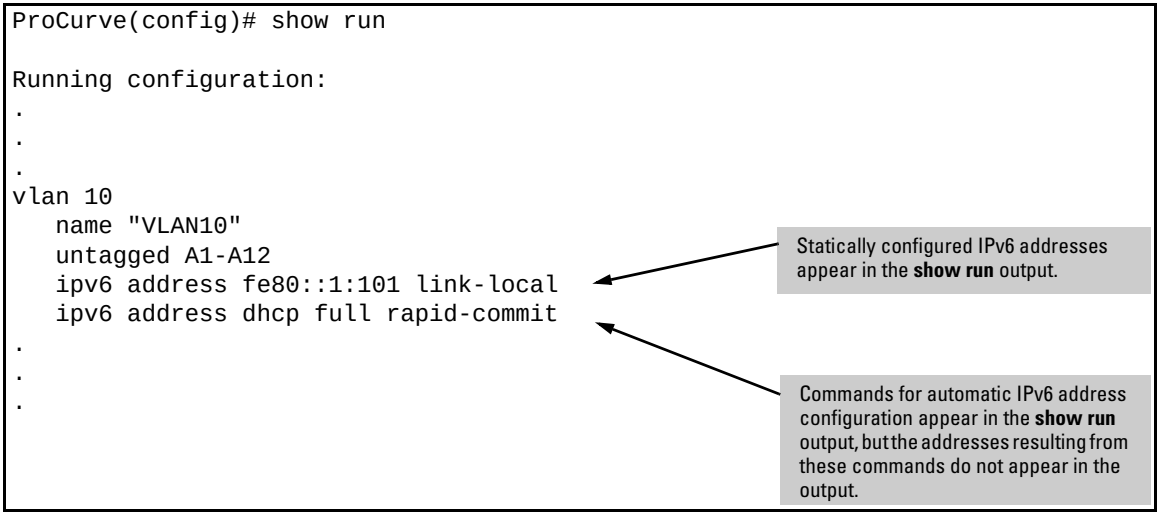
ProCurve(config)# show run
Running configuration:
.
.
.
vlan 10
name "VLAN10"
untagged A1-A12
ipv6 address fe80::1:101 link-local
ipv6 address dhcp full rapid-commit
.
.
.
Statically configured IPv6 addresses appear in the **show run** output.
Commands for automatic IPv6 address configuration appear in the **show run** output, but the addresses resulting from these commands do not appear in the output.

Figure 4-4. Example of Show Run Output Listing the Current IPv6 Addressing Commands

Router Access and Default Router Selection

Routing traffic between destinations on different VLANs configured on the switch or to a destination on an off-switch VLAN is done by placing the switch on the same VLAN interface or subnet as an IPv6-capable router configured to route traffic to other IPv6 interfaces or to tunnel IPv6 traffic across an IPv4 network.

Router Advertisements

An IPv6 router periodically transmits router advertisements (RAs) on the VLANs to which it belongs to notify other devices of its presence. The switch uses these advertisements for purposes such as:

- learning the MAC and link-local addresses of IPv6 routers on the VLAN (For devices other than routers, the switch must use neighbor discovery to learn these addresses.)
- building a list of default (reachable) routers, along with router lifetime and prefix lifetime data
- learning the prefixes and the valid and preferred lifetimes to use for stateless (autoconfigured) global unicast addresses (This is required for autoconfiguration of global unicast IPv6 addresses.)
- learning the hop limit for traffic leaving the VLAN interface
- learning the MTU (Maximum Transmission Unit) to apply to frames intended to be routed

Router Solicitations

When an IPv6 interface becomes operational on the switch, a router solicitation is automatically sent to trigger a router advertisement (RA) from any IPv6 routers reachable on the VLAN. (Router solicitations are sent to the All-Routers multicast address; ff02::2. Refer to “Multicast Application to IPv6 Addressing” on page 3-21.) If an RA is not received within one second of sending the initial router solicitation, the switch sends up to three additional solicitations at intervals of four seconds. If an RA is received, the sending router is added to the switch's default router list and the switch stops sending router solicitations. If an RA is not received, then IPv6 traffic on that VLAN cannot be routed, and the only usable unicast IPv6 address on the VLAN is the link-local address.

Note If the switch does not receive a router advertisement after sending the router solicitations, as described above, then no further router solicitations are sent on that VLAN unless a new IPv6 setting is configured, IPv6 on the VLAN is disabled, then re-enabled, or the VLAN itself is disconnected, then reconnected.

Default IPv6 Router

If IPv6 is enabled on a VLAN where there is at least one accessible IPv6 router, the switch selects a default IPv6 router. (Refer to “Enabling Autoconfiguration of a Global Unicast Address and a Default Router Identity on a VLAN” on page 4-7.)

- If the switch receives router advertisements (RAs) from a single IPv6 router on the same VLAN or subnet, the switch configures a global unicast address and selects the advertising router as the default IPv6 router.
- If multiple IPv6 routers on a VLAN send RAs advertising the same network, the switch configures one global unicast address and selects one router as the default router, based on the router's relative reachability, using factors such as router priority and route cost.
- If multiple IPv6 routers on a VLAN send RAs advertising different subnets, the switch configures a corresponding global unicast address for each RA and selects one of the routers as the default IPv6 router, based on route cost. When multiple RAs are received on a VLAN, the switch uses the router priority and route cost information included in the RAs to identify the default router for the VLAN.

Router Redirection

With multiple routers on a VLAN, if the default (first-hop) router for an IPv6-enabled VLAN on the switch determines that there is a better first-hop router for reaching a given, remote destination, the default router can redirect the switch to use that other router as the default router. For further information on routing IPv6 traffic, refer to the documentation provided for the IPv6 router.

For related information:

- RFC 2461: “Neighbor Discovery for IP Version 6”

View IPv6 Gateway, Route, and Router Neighbors

Use these commands to view the switch's current routing table content and connectivity to routers per VLAN. This includes information received in router advertisements from IPv6 routers on VLANs enabled with IPv6 on the switch.

Viewing Gateway and IPv6 Route Information

Syntax: show ipv6 route [*ipv6-addr*] [connected]

This command displays the routes in the switch's IPv6 routing table.

ipv6-addr*. Optional. Limits the output to show the gateway to the specified IPv6 address.*

connected: *Optional. Limits the output to show only the gateways to IPv6 addresses connected to VLAN interfaces configured on the switch, including the loopback (::1/128) address.*

Dest: *The destination address for a detected route.*

Gateway: *The IPv6 address or VLAN interface used to reach the destination. (Includes the loopback address.)*

Type: *Indicates route type (static, connected, RIP, or OSPF).*

Distance: *The route's administrative distance, used to determine the best path to the destination.*

Metric: *Indicates the route cost for the selected destination.*

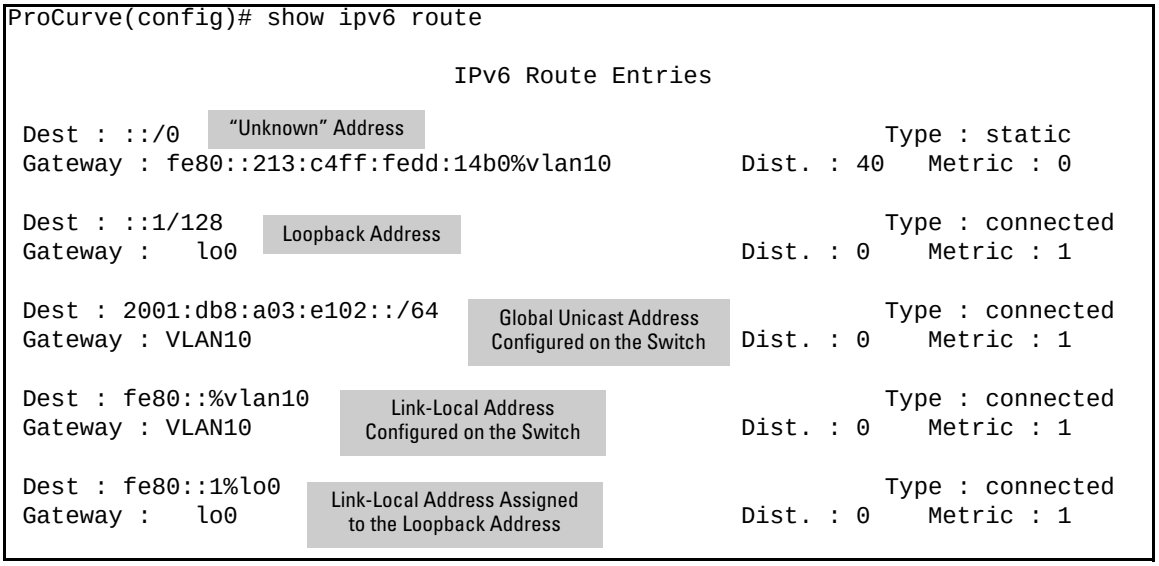

Figure 4-5. Example of Show IPv6 Route Output

Viewing IPv6 Router Information

Syntax: show ipv6 routers [vlan < *vid* >]

This command lists the switch's IPv6 router table entries for all VLANs configured on the switch or for a single VLAN. This output provides information about the IPv6 routers from which routing advertisements (RAs) have been received on the switch.

vlan < *vid* >: Optional. Specifies only the information on IPv6 routers on the indicated VLAN.

Router Address: *The IPv6 address of the router interface.*

Preference: *The relative priority of prefix assignments received from the router when prefix assignments are also received on the same switch VLAN interface from other IPv6 routers.*

Interface: *The VLAN interface on which the path to the router exists.*

MTU: *This is the Maximum Transmission Unit (in bytes) allowed for frames on the path to the indicated router.*

Hop Limit: *The maximum number of router hops allowed.*

Prefix Advertised: *Lists the prefix and prefix size (number of leftmost bits in an address) originating with the indicated router.*

Valid Lifetime: *The total time the address is available, including the preferred lifetime and the additional time (if any) allowed for the address to exist in the deprecated state. Refer to "Address Lifetimes" on page 4-34.*

Preferred Lifetime: *The length of time during which the address can be used freely as both a source and a destination address for traffic exchanges with other devices. Refer to "Address Lifetimes" on page 4-34.*

On/Off Link: Indicates whether the entry source is on the same VLAN as is indicated in the **Interface** field.

For example, figure 4-6 indicates that the switch is receiving router advertisements from a single router that exists on VLAN 10.

```
ProCurve(config)# show ipv6 routers

 IPv6 Router Table Entries

  Router Address : fe80::213:c4ff:fedd:14b0
  Preference     : Medium
  Interface      : VLAN10
  MTU            : 1500
  Hop Limit      : 64

                                                Valid        Preferred    On/Off
  Prefix Advertised                             Lifetime(s)  Lifetime(s)  Link
  ------------------------------------------- ------------ ------------ -------
  2001:db8:a03:e102::/64                      864000       604800       Onlink
```

Figure 4-6. Example of Show IPv6 Routers Output

Address Lifetimes

Every configured IPv6 unicast and anycast address has a lifetime setting that determines how long the address can be used before it must be refreshed or replaced. Some addresses are set as “permanent” and do not expire. Others have both a “preferred” and a “valid” lifetime that specify the duration of their use and availability.

Preferred Lifetime

This is the length of time during which the address can be used freely as both a source and a destination address for traffic exchanges with other devices. This time span is equal to or less than the valid lifetime also assigned to the address. If this time expires without the address being refreshed, the address becomes deprecated and should be replaced with a new, preferred address. In the deprecated state, an address can continue to be used as a destination for existing communication exchanges, but is not used for new exchanges or as a source for traffic sent from the interface. A new, preferred address and its deprecated counterpart will both appear in the **show ipv6 vlan < *vid* >** output as long as the deprecated address is within its valid lifetime.

Valid Lifetime

This is the total time the address is available, and is equal to or greater than the preferred lifetime. The valid lifetime enables communication to continue for transactions that began before the address became deprecated. However, in this timeframe, the address should no longer be used for new communications. If this time expires without the deprecated address being refreshed, the address becomes invalid and may be assigned to another interface.

Sources of IPv6 Address Lifetimes

Manually configured addresses have permanent lifetimes. The prefixes received from router advertisements for global unicast addresses include finite valid and preferred lifetime assignments. Refer to “Unicast Address Prefixes” on page 3-11.

Table 4-1. IPv6 Unicast Addresses Lifetimes

Address Source	Lifetime Criteria
Link-Local	Permanent
Statically Configured Unicast or Anycast	Permanent
Autoconfigured Global	Finite Preferred and Valid Lifetimes
DHCPv6-Configured	Finite Preferred and Valid Lifetimes

A new, preferred address used as a replacement for a deprecated address can be acquired from a manual, DHCPv6, or autoconfiguration source.

5

IPv6 Management Features

Contents

Introduction 5-2

Viewing and Clearing the IPv6 Neighbors Cache 5-2

Viewing the Neighbor Cache 5-3

Clearing the Neighbor Cache 5-5

IPv6 Telnet Operation 5-6

Outbound Telnet to Another Device 5-6

Viewing the Current Telnet Activity on a Switch 5-7

Enabling or Disabling Inbound Telnet Access 5-8

Viewing the Current Inbound Telnet Configuration 5-8

SNTP and Timep 5-9

Configuring (Enabling or Disabling) the SNTP Mode 5-9

Configuring an IPv6 Address for an SNTP Server 5-10

Configuring (Enabling or Disabling) the Timep Mode 5-12

TFTP File Transfers Over IPv6 5-15

TFTP File Transfers over IPv6 5-15

Enabling TFTP for IPv6 5-16

Using TFTP to Copy Files over IPv6 5-17

Using Auto-TFTP for IPv6 5-20

SNMP Management for IPv6 5-21

SNMP Features Supported 5-21

SNMP Configuration Commands Supported 5-22

SNMPv1 and V2c 5-22

SNMPv3 5-22

IP Preserve for IPv6 5-24

Introduction

Feature	Default	CLI
Neighbor Cache	n/a	5-3, 5-5
Telnet6	Enabled	5-6, 5-7, 5-8
SNTP Address	None	5-10
Timep Address	None	5-13
TFTP	n/a	5-15
SNMP Trap Receivers	None	5-22

This chapter focuses on the IPv6 application of management features that support both IPv6 and IPv4 operation. For additional information on these features, refer to the current *Management and Configuration Guide* for your switch.

Viewing and Clearing the IPv6 Neighbors Cache

Neighbor discovery occurs when there is communication between the switch and another, reachable IPv6 device on the same VLAN. A neighbor destination is reachable from a given source address if a confirmation (neighbor solicitation) has been received at the source verifying that traffic has been received at the destination.

The switch maintains an IPv6 neighbor cache that is populated as a result of communication with other devices on the same VLAN. You can view and clear the contents of the neighbor cache using the commands described in this section.

Anycast Addresses. Multiple, duplicate addresses configured as Anycast on different devices are special cases of unicast addresses and are not identified as duplicates by the Neighbor Discovery process. Refer to “Anycast Addresses” on page 3-20.

Viewing the Neighbor Cache

Neighbor discovery occurs when there is communication between IPv6 devices on a VLAN. The Neighbor Cache retains data for a given neighbor until the entry times out. For more on this topic, refer to "Neighbor Discovery (ND)" on page 4-17.

Syntax: show ipv6 neighbors [vlan < *vid* >]

Displays IPv6 neighbor information currently held in the neighbor cache. After a period without communication with a given neighbor, the switch drops that neighbor's data from the cache. The command lists neighbors for all VLAN interfaces on the switch or for only the specified VLAN. The following fields are included for each entry in the cache:

IPv6 Address: *Lists the 128-bit addresses for the local host and any neighbors (on the same VLAN) with whom there has been recent communication.*

MAC Address: *The MAC Address corresponding to each of the listed IPv6 addresses.*

VLAN < *vid* >: *Optional. Causes the switch to list only the IPv6 neighbors on a specific VLAN configured on the switch.*

Type: *Appears only when VLAN is not specified, and indicates whether the corresponding address is* ***local*** *(configured on the switch) or* **dynamic** *(configured on a neighbor device).*

Age: *Appears only when VLAN is specified, and indicates the length of time the entry has remained unused.*

Port: *Identifies the switch port on which the entry was learned. If this field is empty for a given address, then the address is configured on the switch itself.*

State: *A neighbor destination is reachable from a given source address if confirmation has been received at the source verifying that traffic has been received at the destination. This field shows the reachability status of each listed address:*

- **INCMP** (*Incomplete*)**:** *Neighbor address resolution is in progress, but has not yet been determined.*
- **REACH** (*Reachable*)**:** *The neighbor is known to have been reachable recently.*

— Continued on the next page. —

— Continued from previous page. —

- **STALE:** *A timeout has occurred for reachability of the neighbor, and an unsolicited discovery packet has been received from the neighbor address. If the path to the neighbor is then used successfully, this state is restored to* **REACH**.
- **DELAY:** *Indicates waiting for a response to traffic sent recently to the neighbor address. The time period for determining the neighbor's reachability has been extended.*
- **PROBE:** *The neighbor may not be reachable. Periodic, unicast neighbor solicitations are being sent to verify reachability.*

```
ProCurve(config)# show ipv6 neighbor

 IPv6 ND Cache Entries

 IPv6 Address                            MAC Address   State Type    Port
 --------------------------------------- ------------- ----- ------- ----
 2001:db8:260:212::101                   0013c4-dd14b0 STALE dynamic A1
 2001:db8:260:214::1:15                  001279-88a100 REACH local
 fe80::1:1                               001279-88a100 REACH local
 fe80::10:27                             001560-7aadc0 REACH dynamic A3
 fe80::213:c4ff:fedd:14b0                0013c4-dd14b0 REACH dynamic A1
```

Figure 5-1. Example of Neighbor Cache Without Specifying a VLAN

```
ProCurve(config)# show ipv6 neighbor vlan 10

 IPv6 ND Cache Entries

 IPv6 Address                            MAC Address   State Age           Port
 --------------------------------------- ------------- ----- ------------- ----
 2001:db8:260:212::101                   0013c4-dd14b0 STALE 5h:13m:44s    A1
 2001:db8:260:214::1:15                  001279-88a100 REACH 11h:15m:23s   B17
 fe80:1a3::1:1                           001279-88a100 REACH 9h:35m:11s    B12
 fe80:::10:27                            001560-7aadc0 REACH 22h:26m:12s   A3
 fe80::213:c4ff:fedd:14b0                0013c4-dd14b0 REACH 23 0h:32m:36s A1
```

Figure 5-2. Example of Neighbor Cache Content for a Specific VLAN

Clearing the Neighbor Cache

When there is an event such as a topology change or an address change, the neighbor cache may have too many entries to allow efficient use. Also, if an unauthorized client is answering DAD or normal neighbor solicitations with invalid replies, the neighbor cache may contain a large number of invalid entries and communication with some valid hosts may fail and/or the **show ipv6 neighbors** command output may become too cluttered to efficiently read. In such cases, the fastest way to restore optimum traffic movement on a VLAN may be to statically clear the neighbor table instead of waiting for the unwanted entries to time-out.

Syntax: clear ipv6 neighbors

Syntax:*Executed at the global config level, this command removes all non-local IPv6 neighbor addresses and corresponding MAC addresses from the neighbor cache except neighbor entries specified as next-hops for active routes. Note that the Layer-2 address information for any next-hop route is cleared until the route is refreshed in the neighbor cache.*

```
ProCurve(config)# clear ipv6 neighbors
ProCurve(config)# show ipv6 neighbors

ProCurve# show ipv6 neighbors

 IPv6 ND Cache Entries

 IPv6 Address                            MAC Address   State Type    Port
 --------------------------------------- ------------- ----- ------- ----
 fe80::213:c4ff:fedd:14b0                000000-000000 INCMP dynamic
```

For an active-route next-hop, the MAC address and source port data is removed, and the State is set to "Incomplete" until the route is refreshed in the neighbor cache.

Figure 5-3. Example of Clearing the IPv6 Neighbors Cache

IPv6 Telnet Operation

This section describes Telnet operation for IPv6 on the switch. For IPv4 Telnet operation, refer to the *Management and Configuration Guide* for your switch.

Outbound Telnet to Another Device

Syntax: telnet < *link-local-addr* >%vlan< *vid* >
telnet < *global-unicast-addr* >

Outbound Telnet establishes a Telnet session from the switch CLI to another IPv6 device, and includes these options.

- *Telnet for Link-Local Addresses on the same VLAN requires the link-local address and and interface scope:*

  < *link-local-addr* >: *Specifies the link-local IPv6 address of the destination device.*

  %vlan< *vid* >: *Suffix specifying the interface on which the destination device is located. No spaces are allowed in the suffix.*

- *Telnet for Global Unicast Addresses requires a global unicast address for the destination. Also, the switch must be receiving router advertisements from an IPv6 gateway router.*

  < *global-unicast-addr* >: *Specifies the global IPv6 address of the destination device.*

For example, to Telnet to another IPv6 device having a link-local address of fe80::215:60ff:fe79:8980 and on the same VLAN interface (VLAN 10), you would use the following command:

```
ProCurve(config)# telnet fe80::215:60ff:fe79:980%vlan10
```

If the switch is receiving router advertisements from an IPv6 default gateway router, you can Telnet to a device on the same VLAN or another VLAN or subnet by using its global unicast address. For example, to Telnet to a device having an IPv6 global unicast address of 2001:db8::215:60ff:fe79:980, you would enter the following command:

```
ProCurve(config)# telnet 2001:db8::215:60ff:fe79:980
```

Viewing the Current Telnet Activity on a Switch

Syntax: show telnet

This command shows the active incoming and outgoing telnet sessions on the switch (for both IPv4 and IPv6). Command output includes the following:

Session: *The session number. The switch allows one outbound session and up to five inbound sessions.*

Privilege: *Manager or Operator.*

From: *Console (for outbound sessions) or the source IP address of the inbound session.*

To: *The destination of the outbound session, if in use.*

For example, the following figure shows that the switch is running one outbound, IPv4 session and is being accessed by two inbound sessions.

```
ProCurve# show telnet

 Telnet Activity

 ----------------------------------------------------------
 Session  :     1
 Privilege: Manager
 From     : Console
 To       : 10.0.10.140
 ----------------------------------------------------------
 Session  :     2
 Privilege: Manager
 From     : 2620:0:260:212::2:219
 To       :
 ----------------------------------------------------------
 Session  : **  3
 Privilege: Manager
 From     : fe80::2:101
 To       :
```

The ** in the "Session: indicates the session through which **show telnet** was run.

Figure 5-4. Example of Show Telnet Output with Three Sessions Active

Enabling or Disabling Inbound Telnet Access

Syntax: [no] telnet-server

This command is used at the global config level to enable (the default) or disable all (IPv4 and IPv6) inbound Telnet access to the switch.

The **no** *form of the command disables inbound telnet.*

For example, to disable IPv4 and IPv6 Telnet access to the switch, you would use this command:

```
ProCurve(config)# no telnet-server
```

Viewing the Current Inbound Telnet Configuration

Syntax: show console

This command shows the current configuration of IPv4 and IPv6 inbound telnet permissions, as well as other information. For both protocols, the default setting allows inbound sessions.

```
LPE-5400-a100(config)# show console

 Console/Serial Link

  Inbound Telnet Enabled [Yes] : Yes
  Web Agent Enabled [Yes] : Yes
  Terminal Type [VT100] : VT100
  Screen Refresh Interval (sec) [3] : 3
  Displayed Events [All] : All

  Baud Rate [Speed Sense] : speed-sense
  Flow Control [XON/XOFF] : XON/XOFF
  Session Inactivity Time (min) [0] : 0
```

Inbound Telnet Setting for IPv4 and IPv6 Telnet

Figure 5-5. Show Console Output Showing Default Console Configuration

SNTP and Timep

Configuring (Enabling or Disabling) the SNTP Mode

Software release K.13.01 and greater enables configuration of a global unicast address for IPv6 SNTP time server.

This section lists the SNTP and related commands, including an example of using an IPv6 address. For the details of configuring SNTP on the switch, refer to the chapter titled "Time Protocols" in the *Management and Configuration Guide* for your switch.

The following commands are available at the global config level for SNTP operation.

Commands Affecting SNTP	Function
show sntp	Display the current SNTP configuration.
timesync < sntp | timep >	Enable either SNTP or Timep as the time synchronization method on the switch without affecting the configuration of either.
[no] timesync	Enable time synchronization. (Requires a timesync method to also be enabled.) The no version disable time synchronization without affecting the configuration of the current time synchronization method.)
[no]sntp	Enables SNTP with the current SNTP configuration. The no version disables SNTP without changing the current SNTP configuration.
sntp < unicast | broadcast >	Configures the SNTP mode. (Default: Broadcast)
sntp < 30 - 720 >	Changes the interval between time requests. (Default: 720 seconds)

Configuring an IPv6 Address for an SNTP Server

Note

To use a global unicast IPv6 address to configure an IPv6 SNTP time server on the switch, the switch must be receiving advertisements from an IPv6 router on a VLAN configured on the switch.

To use a link-local IPv6 address to configure an IPv6 SNTP time server on the switch, it is necessary to append **%vlan** followed immediately (without spaces) by the VLAN ID of the VLAN on which the server address is available. (The VLAN must be configured on the switch.) For example:

```
fe80::11:215%vlan10
```

Syntax:. [no] sntp server priority < 1 - 3 > < *link-local-addr* >%vlan< *vid* > [1 - 7]
[no] sntp server priority < 1 - 3 > < *global-unicast-addr* > [1 - 7]

Configures an IPv6 address for an SNTP server.

server priority < 1 - 3 >: *Specifies the priority of the server addressing being configured. When the SNTP mode is set to unicast and more than one server is configured, this value determines the order in which the configured servers will be accessed for a time value. The switch polls multiple servers in order until a response is received or all servers on the list have been tried without success. Up to three server addresses (IPv6 and/or IPv4) can be configured.*

< link-local-addr >: *Specifies the link-local IPv6 address of the destination device.*

%vlan< *vid* >: *Suffix specifying the interface on which the destination device is located. No spaces are allowed in the suffix.*

< global-unicast-addr >: *Specifies the global IPv6 address of the destination device.*

[1 - 7]: *This optional setting specifies the SNTP server version expected for the specified server. (Default: 3)*

For example, to configure link-local and global unicast SNTP server addresses of:

- fe80::215:60ff:fe7a:adc0 (on VLAN 10, configured on the switch)
- 2001:db8::215:60ff:fe79:8980

as the priority "1" and "2" SNTP servers, respectively, using version 7, you would enter these commands at the global config level, as shown below.

```
ProCurve(config)# sntp server priority 1
fe80::215:60ff:fe7a:adc0%vlan10 7

ProCurve(config)# sntp server priority 2
2001:db8::215:60ff:fe79:8980 7
```

Note

In the preceding example, using a link-local address requires that you specify the local scope for the address; VLAN 10 in this case. This is always indicated by **%vlan** followed immediately (without spaces) by the VLAN identifier.

Syntax:. show sntp

Displays the current SNTP configuration, including the following:

Time Sync Mode: *Indicates whether timesync is disabled or set to either SNTP or Timep. (Default:* **timep***)*

SNTP Mode: *Indicates whether SNTP uses the broadcast or unicast method of contacting a time server. The broadcast option does not require you to configure a time server address. The unicast option does require configuration of a time server address.*

Poll Interval: *Indicates the interval between consecutive time requests to an SNTP server.*

Priority: *Indicates the configured priority for the corresponding SNTP server address.*

SNTP Server Address: *Lists the currently configured SNTP server addresses.*

Protocol Version: *Lists the SNTP server protocol version to expect from the server at the corresponding address.*

For example, the **show sntp** output for the proceeding **sntp server** command example would appear as follows:

```
ProCurve(config)# show sntp

 SNTP Configuration

  Time Sync Mode: Sntp
  SNTP Mode : Broadcast
  Poll Interval (sec) [720] : 719

  Priority SNTP Server Address                          Protocol Version
  -------- ------------------------------------------- ----------------
  1        2001:db8::215:60ff:fe79:8980                 7
  2        10.255.5.24                                  3
```

This example illustrates the command output when both IPv6 and IPv4 server addresses are configured.

Figure 5-6. Example of Show SNTP Output with Both an IPv6 and an IPv4 Server Address Configured

Note that the **show management** command can also be used to display SNTP server information.

Configuring (Enabling or Disabling) the Timep Mode

Software release K.13.01 and greater enables configuration of a global unicast address for IPv6 Timep time server.

This section lists the Timep and related commands, including an example of using an IPv6 address. For the details of configuring Timep on the switch, refer to the chapter titled "Time Protocols" in the *Management and Configuration Guide* for your switch.

The following commands are available at the global config level for Timep operation.

Commands Affecting Timep	Function
show timep	Display the current timep configuration.
timesync < sntp | timep >	Enable either SNTP or Timep as the time synchronization method on the switch without affecting the configuration of either.
ip timep dhcp [interval] < 1 - 9999 >]	Enable Timep operation with a Timep server assignment configured from an IPv4 or IPv6 DHCP server. Optionally change the interval between time requests.

ip timep manual < *ipv6-addr* > [interval < 1 - 9999 >]	Enable Timep operation with a statically configured IPv6 address for a Timep server. Optionally change the interval between time requests.
no ip timep	Disables Timep operation. To re-enable Timep, it is necessary to reconfigure either the DHCP or the static option.

Note

To use a global unicast IPv6 address to configure an IPv6 Timep server on the switch, the switch must be receiving advertisements from an IPv6 router on a VLAN configured on the switch.

To use a link-local IPv6 address to configure an IPv6 Timep server on the switch, it is necessary to append **%vlan** followed (without spaces) by the VLAN ID of the VLAN on which the server address is available. The VLAN must be configured on the switch. For example: `fe80::11:215%vlan10`

Syntax:. ip timep dhcp [interval < 1 - 9999 >]
ip timep manual < *ipv6-addr* | *ipv4-addr* > [interval < 1 - 9999 >]

Used at the global config level to configure a Timep server address.

Note: *The switch allows one Timep server configuration.*

timep dhcp: *Configures the switch to obtain the address of a Timep server from an IPv4 or IPv6 DHCP server.*

timep manual: *Specifies static configuration of a Timep server address.*

< *ipv6-addr* >: *Specifies the IPv6 address of an SNTP server. Refer to preceding* **Note**.

[Interval < 1 - 9999 >]: *This optional setting specifies the interval in minutes between Timep requests. (Default: 720)*

For example, to configure a link-local Timep server address of:

fe80::215:60ff:fe7a:adc0

where the address is on VLAN 10, configured on the switch, you would enter this command at the global config level, as shown below.

```
ProCurve(config)# ip timep manual
fe80::215:60ff:fe7a:adc0%vlan10
```

Note

In the preceding example, using a link-local address requires that you specify the local scope for the address; VLAN 10 in this case. This is always indicated by **%vlan** followed immediately (without spaces) by the VLAN identifier. For a global unicast address, you would enter the address *without* the **%vlan** suffix.

Syntax:. show timep

Displays the current Timep configuration, including the following:

Time Sync Mode: *Indicates whether timesync is disabled or set to either SNTP or Timep. (Default: Disabled)*

Timep Mode: *Indicates whether Timep is configured to use a DHCP server to acquire a Timep server address or to use a statically configured Timep server address.*

Server Address: *Lists the currently configured Timep server address.*

Poll Interval (min) [720]: *Indicates the interval between consecutive time requests to the configured Timep server.*

For example, the **show timep** output for the preceding **ip timep manual** command example would appear as follows:

```
ProCurve(config)# sho timep

 Timep Configuration

  Time Sync Mode: Timep
  TimeP Mode [Disabled] : Manual
  Server Address : fe80::215:60ff:fe7a:adc0%vlan10
  Poll Interval (min) [720] : 720
```

Figure 5-7. Example of Show Timep Output with an IPv6 Server Address Configured

Note that the **show management** command can also be used to display Timep server information.

TFTP File Transfers Over IPv6

You can use TFTP **copy** commands over IPv6 to upload, or download files to and from a physically connected device or a remote TFTP server, including:

- Switch software
- Software images
- Switch configurations
- ACL command files
- Diagnostic data (crash data, crash log, and event log)

For complete information on how to configure TFTP file transfers between the switch and a TFTP server or other host device on the network, refer to the "File Transfers" appendix in the *Management and Configuration Guide* for your switch.

To upload and/or download files to the switch using TFTP in an IPv6 network, you must:

1. Enable TFTP for IPv6 on the switch (see "Enabling TFTP for IPv6" on page 5-16).
2. Enter a TFTP **copy** command with the IPv6 address of a TFTP server in the command syntax (see "Using TFTP to Copy Files over IPv6" on page 5-17).
3. (Optional) To enable auto-TFTP operation, enter the **auto-tftp** command (see "Using Auto-TFTP for IPv6" on page 5-20).

Enabling TFTP for IPv6

Client and server TFTP for IPv6 is enabled by default on the switch. However, if it is disabled, you can re-enable it by specifying TFTP client or server functionality with the **tftp** <**client** | **server**> command. Enter the **tftp** < **client** | **server**> command at the global configuration level.

Syntax: [no] tftp <client | server>

Enables TFTP for IPv4 and IPv6 client or server functionality so that the switch can:

- *Use TFTP client functionality to access IPv4- or IPv6-based TFTP servers in the network to receive downloaded files.*
- *Use TFTP server functionality on the switch to be accessed by other IPv4 or IPv6 hosts requesting to upload files.*

The **no** *form of the command disables the client or server functionality.*

Default: TFTP Client and Server functionality enabled

Usage Notes

To disable all TFTP client or server operation on the switch except for the auto-TFTP feature, enter the **no tftp** <**client** | **server**> command. To re-enable TFTP client or server operation, re-enter the **tftp** <**client** | **server**> command. (Entering **no tftp** without specifying client or server affects only the client functionality. To disable or re-enable the TFTP server functionality, you must specify **server** in the command.)

When TFTP is disabled, instances of TFTP in the CLI **copy** command and the Menu interface “Download OS” screen become unavailable.

The **[no] tftp** <**client** | **server**> command does not affect auto-TFTP operation. For more information, see “Using Auto-TFTP for IPv6” on page 5-20.

Using TFTP to Copy Files over IPv6

Use the TFTP **copy** commands described in this section to:

- Download specified files from a TFTP server to a switch on which TFTP client functionality is enabled.
- Upload specified files from a switch, on which TFTP server functionality is enabled, to a TFTP server.

Syntax: copy tftp < *target* > < *ipv6-addr* > < *filename* >

Copies (downloads) a data file from a TFTP server at the specified IPv6 address to a target file on a switch that is enabled with TFTP server functionality.

< *ipv6-addr* >: *If this is a link-local address, use this IPv6 address format:*

fe80::< *device-id* >%vlan< *vid* >

For example: `fe80::123%vlan10`

If this is a global unicast or anycast address, use this IPv6 format:

< *ipv6-addr* >

For example: `2001:db8::123`

< *target* > *is one of the following values:*

- **autorun-cert-file**: Copies an autorun trusted certificate to the switch.
- **autorun-key-file**: Copies an autorun key file to the switch.
- **command-file**: Copies a file stored on a remote host and executes the ACL command script on the switch. Depending on the ACL commands stored in the file, one of the following actions is performed in the running-config file on the switch:
  - *A new ACL is created.*
  - *An existing ACL is replaced.*
  - **match**, **permit**, *or* **deny** *statements are added to an existing ACL.*

  For more information on ACLs, refer to "Creating an ACL Offline" in the Access Control Lists (ACLs) chapter in the Access Security Guide.
- **config < *filename* >:** *Copies the contents of a file on a remote host to a configuration file on the switch.*

- **flash** < **primary** | **secondary** >: *Copies a software file stored on a remote host to primary or secondary flash memory on the switch. To run a newly downloaded software image, enter the* **reload** *or* **boot system flash** *command.*
- **pub-key-file**: *Copies a public-key file to the switch.*
- **startup-config**: *Copies a configuration file on a remote host to the startup configuration file on the switch.*

.

Syntax: copy <*source* > tftp < *ipv6-addr* > < *filename* > < pc | unix >

Copies (uploads) a source data file on a switch that is enabled with TFTP server functionality to a file on the TFTP server at the specified IPv6 address, where <source> is one of the following values:

- **command-output < cli-command >**: *Copies the output of a CLI command to the specified file on a remote host.*
- **config < *filename* >:** *Copies the specified configuration file to a remote file on a TFTP server.*
- **crash-data < slot-id | master >:** *Copies the contents of the crash data file to the specified file path on a remote host. The crash data is software-specific and used to determine the cause of a system crash. You can copy crash information from an individual slot or from the master crash file on the switch.*
- **crash-log < *slot-id* | master >:** *Copies the contents of the crash log to the specified file path on a remote host. The crash log contains processor-specific operational data that is used to determine the cause of a system crash. You can copy the contents of the crash log from an individual slot or from the master crash log on the switch.*
- **event-log**: *Copies the contents of the Event Log on the switch to the specified file path on a remote host.*
- **flash < primary | secondary >:** *Copies the software file used as the primary or secondary flash image on the switch to a file on a remote host.*
- **startup-config**: *Copies the startup configuration file in flash memory to a remote file on a TFTP server.*
- **running-config**: *Copies the running configuration file to a remote file on a TFTP server.*

< ipv6-addr >: *If this is a link-local address, use this IPv6 address format:*

fe80::< *device-id* >%vlan< *vid* >

For example: `fe80::123%vlan10`

If this is a global unicast or anycast address, use this IPv6 format:

< *ipv6-addr* >

For example: `2001:db8::123`

Using Auto-TFTP for IPv6

At switch startup, the auto-TFTP for IPv6 feature automatically downloads a software image to the switch from a specified TFTP server, then reboots the switch. To implement the process the switch must first reboot using one of the following methods:

- enter the **boot system flash primary** command in the CLI
- with the default flash boot image set to primary flash (the default), enter the **boot** or the **reload** command, or cycle the power to the switch. (To reset the boot image to primary flash, use **boot set-default flash primary**.)

Syntax: auto-tftp <*ipv6-addr*> <*filename*>

Configures the switch to automatically download the specified software file from the TFTP server at the specified IPv6 address. The file is downloaded into primary flash memory at switch startup. The switch then automatically reboots from primary flash.

Notes: *To enable auto-TFTP to copy a software image to primary flash memory, the version number of the downloaded software file (for example, K_14_01.swi) must be different from the version number currently in the primary flash image.*

The current TFTP client status (enabled or disabled) does not affect auto-TFTP operation. (Refer to "Enabling TFTP for IPv6" on page 5-16.)

Completion of the auto-TFTP process may require several minutes while the switch executes the TFTP transfer to primary flash, and then reboots again.

The **no** *form of the command disables auto-TFTP operation by deleting the* **auto-tftp** *entry from the startup configuration. The* **no auto-tftp** *command does not affect the current TFTP-enabled configuration on the switch. However, entering the* **ip ssh filetransfer** *command automatically disables both* **auto-tftp** *and* **tftp** *operation.*

SNMP Management for IPv6

As with SNMP for IPv4, you can manage a switch via SNMP from an IPv6-based network management station by using an application such as ProCurve Manager (PCM) or ProCurve Manager Plus (PCM+). (For more on PCM and PCM+, go to the ProCurve Networking web site at **www.procurve.com**.)

SNMP Features Supported

The same SNMP for IPv4 features are supported over IPv6:

- access to a switch using SNMP version 1, version 2c, or version 3
- enhanced security with the configuration of SNMP communities and SNMPv3 user-specific authentication password and privacy (encryption) settings
- SNMP notifications, including:
  - SNMP version 1 or SNMP version 2c traps
  - SNMPv2c informs
  - SNMPv3 notification process, including traps
- Advanced RMON (Remote Monitoring) management
- ProCurve Manager or ProCurve Manager Plus management applications
- Flow sampling using sFlow
- Standard MIBs, such as the Bridge MIB (RFC 1493) and the Ethernet MAU MIB (RFC 1515)

SNMP Configuration Commands Supported

IPv6 addressing is supported in the following SNMP configuration commands: For more information on each SNMP configuration procedure, refer to the "Configuring for Network Management Applications" chapter in the current *Management and Configuration Guide* for your switch.

SNMPv1 and V2c

Syntax:. snmp-server host < *ipv4-addr* | *ipv6-addr* > < *community-name* > [none | all | non-info | critical | debug] [inform [retries < *count* >] [timeout < *interval* >]]

Executed at the global config level to configure an SNMP trap receiver to receive SNMPv1 and SNMPv2c traps, SNMPv2c informs, and (optionally) event log messages

SNMPv3

Syntax: snmpv3 targetaddress < *name* > params < *parms_name* > <*ipv4-addr* | *ipv6-addr*> [addr-mask < *ip4-addr* >] [filter < none | debug | all | not-info | critical>] [max-msg-size < 484-65535 >] [port-mask < *tcp-udp port* >] [retries < 0 - 255 >] [taglist <*tag_name*>] [timeout < 0 - 2147483647 >] [udp-port *port-number*]

Executed at the global config level to configure an SNMPv3 management station to which notifications (traps and informs) are sent.

Note

IPv6 is not supported in the configuration of an interface IPv6 address as the default source IP address used in the IP headers of SNMP notifications (traps and informs) or responses sent to SNMP requests. Only IPv4 addresses are supported in the following configuration commands:

snmp-server trap-source < *ipv4-addr* | loopback < 0-7 >>

snmp-server response-source [*dst-ip-of-request* | *ipv4-addr* | loopback < 0-7 >]

IPv6 addresses are supported in SNMP **show** command output as shown in Figure 5-8 and Figure 5-9.

The **show snmp-server** command displays the current SNMP policy configuration, including SNMP communities, network security notifications, link-change traps, trap receivers (including the IPv4 or IPv6 address) that can receive SNMPv1 and SNMPv2c traps, and the source IP (interface) address used in IP headers when sending SNMP notifications (traps and informs) or responses to SNMP requests.

```
ProCurve(config)# show snmp-server

 SNMP Communities

  Community Name       MIB View Write Access
  -------------------- -------- ------------
  public               Manager  Unrestricted
  marker               Manager  Unrestricted

 Trap Receivers

  Link-Change Traps Enabled on Ports [All] : All

  Traps Category                   Current Status
  ---------------------------      ---------------
  SNMP Authentication            : Extended
  Password change                : Enabled
  Login failures                 : Enabled
  Port-Security                  : Enabled
  Authorization Server Contact   : Enabled
  DHCP-Snooping                  : Enabled
  Dynamic ARP Protection         : Enabled

  Address                 Community              Events   Type   Retry   Timeout
  ----------------------- ---------------------- -------- ------ ------- -------
  15.29.17.218            public                 All      trap   3       15
  15.29.17.219            public                 Critical trap   3       15
  2620:0000:0260:0211
    :0217:a4ff:feff:1f70  marker                 Critical trap   3       15

 Excluded MIBs


 Snmp Response Pdu Source-IP Information

  Selection Policy   : rfc1517

 Trap Pdu Source-IP Information

  Selection Policy   : rfc1517
```

An IPv6 address is displayed on two lines.

Figure 5-8. "show snmp-server" Command Output with IPv6 Address

The **show snmpv3 targetaddress** command displays the configuration (including the IPv4 or IPv6 address) of the SNMPv3 management stations to which notification messages are sent.

```
ProCurve(config)# show snmpv3 targetaddress

 snmpTargetAddrTable [rfc2573]

  Target Name               IP Address             Parameter
  ------------------------- ---------------------- ---------------------------
  1                         15.29.17.218           1
  2                         15.29.17.219           2
  PP.217                    15.29.17.217           marker_p
  PP.218                    2620:0:260:211
                              :217:a4ff:feff:1f70  marker_p
```

An IPv6 address is displayed on two lines.

Figure 5-9. "show snmpv3 targetaddress" Command Output with IPv6 Address

IP Preserve for IPv6

IPv6 supports the IP Preserve feature, which allows you to copy a configuration file from a TFTP server to multiple switches without overwriting the IPv6 address and subnet mask on VLAN 1 (default VLAN) in each switch, and the Gateway IPv6 address assigned to the switch.

To configure IP Preserve, enter the **ip preserve** statement at the end of the configuration file that will be downloaded from a TFTP server. (Note that you do not invoke IP Preserve by entering a command from the CLI).

```
; J8697A Configuration Editor; Created on release #K.14.01
hostname "ProCurve"
time daylight-time-rule None

   *
   *
   *
   *
   *
   *
password manager
password operator
ip preserve
```

Entering an **ip preserve** statement as the last line in a configuration file stored on a TFTP server allows you to download and execute the file as the startup-config file on an IPv6 switch. When the switch reboots, the configuration settings in the downloaded file are implemented without changing the IPv6 address and gateway assigned to the switch as shown in Figure 5-11.

Figure 5-10. Example of How to Enter IP Preserve in a Configuration File

To download an IP Preserve configuration file to an IPv6-based switch, enter the TFTP **copy** command as described in "You can use TFTP copy commands over IPv6 to upload, or download files to and from a physically connected device or a remote TFTP server, including:" on page 5-15 to copy the file as the new startup-config file on a switch.

When you download an IP Preserve configuration file, the following rules apply:

- If the switch's current IPv6 address for VLAN 1 was statically configured and not dynamically assigned by a DHCP/Bootp server, the switch reboots and retains its current IPv6 address, subnet mask, and gateway address. All other configuration settings in the downloaded configuration file are applied.
- If the switch's current IPv6 address for VLAN 1 was assigned from a DHCP server and not statically configured, IP Preserve is suspended. The IPv6 addressing specified in the downloaded configuration file is implemented when the switch copies the file and reboots.
  - If the downloaded file specifies DHCP/Bootp as the source for the IPv6 address of VLAN 1, the switch uses the IPv6 address assigned by the DHCP/Bootp server.
  - If the file specifies a dedicated IPv6 address and subnet mask for VLAN 1 and a Gateway IPv6 address, the switch implements these settings in the startup-config file.

To verify how IP Preserve was implemented in a switch, after the switch reboots, enter the **show run** command. Figure 5-11 shows an example in which all configurations settings have been copied into the startup-config file except for the IPv6 address of VLAN 1 (2001:db8::214:c2ff:fe4c:e480) and the default IPv6 gateway (2001:db8:0:7::5), which were retained.

Note that if a switch received its IPv6 address from a DHCP server, the "ip address" field under "vlan 1" would display: **dhcp-bootp**.

```
ProCurve(config)# show run

Running configuration:

; J8715A Configuration Editor; Created on release #K.14.01

hostname "ProCurve"
module 1 type J8702A
module 2 type J8705A
trunk A11-A12 Trk1 Trunk
ip default-gateway 2001:db8:0:7::5
snmp-server community "public" Unrestricted
vlan 1
   name "DEFAULT_VLAN"
   untagged A1-A10,A13-A24,B1-B24,Trk1
   ip address 2001:db8::214:c2ff:fe4c:e480
   exit
spanning-tree Trk1 priority 4
password manager
password operator
```

Because the switch's IPv6 address and default gateway were statically configured (not assigned by a DHCP server), when the switch boots up with the IP Preserve startup configuration file (see Figure 5-10), its current IPv6 address, subnet mask, and default gateway are not changed.

If a switch's current IP address was acquired from a DHCP/Bootp server, the IP Preserve statement is ignored and the IP addresses in the downloaded configuration file are implemented.

Figure 5-11. Configuration File with Dedicated IP Addressing After Startup with IP Preserve

For more information on how to use the IP Preserve feature, refer to the "Configuring IP Addressing" chapter in the current *Management and Configuration Guide* for your ProCurve switch.

6

IPv6 Management Security Features

Contents

IPv6 Management Security 6-2

Authorized IP Managers for IPv6 6-3

Usage Notes 6-3

Configuring Authorized IP Managers for Switch Access 6-5

Using a Mask to Configure Authorized Management Stations 6-5

Configuring Single Station Access 6-5

Configuring Multiple Station Access 6-6

Displaying an Authorized IP Managers Configuration 6-12

Additional Examples of Authorized IPv6 Managers Configuration . 6-13

Secure Shell (SSH) for IPv6 6-15

Configuring SSH for IPv6 6-15

Displaying an SSH Configuration 6-18

Secure Copy and Secure FTP for IPv6 6-19

IPv6 Management Security

This chapter describes management security features that are IPv6 counterparts of IPv4 management security features on the switches covered by this guide.

Feature	Default	CLI
configure authorized IP managers for IPv6	disabled	6-5
configuring secure shell for IPv6	disabled	6-15
enabling secure copy and secure FTP for IPv6	disabled	6-19

This chapter describes the following IPv6-enabled management security features:

- Authorized IP Managers for IPv6
- Secure Shell for IPv6
- Secure Copy and Secure FTP for IPv6

Authorized IP Managers for IPv6

The Authorized IP Managers feature uses IP addresses and masks to determine which stations (PCs or workstations) can access the switch through the network. This feature supports switch access through:

- Telnet and other terminal emulation applications
- Web browser interface
- SNMP (with a correct community name)
- SSH
- TFTP

As with the configuration of IPv4 management stations, the Authorized IP Managers for IPv6 feature allows you to specify the IPv6-based stations that can access the switch.

Usage Notes

- You can configure up to 100 authorized IPv4 and IPv6 manager *addresses* on a switch, where each address applies to either a single management station or a group of stations. Each authorized manager address consists of an IPv4 or IPv6 address and a mask that determines the individual management stations that are allowed access.
  - You configure authorized IPv4 manager addresses using the **ip authorized-managers** command. For more information, refer to the "Using Authorized IP Managers" chapter in the *Access Security Guide*.
  - You configure authorized IPv6 manager addresses using the **ipv6 authorized-managers** command. For more information, see "Configuring Authorized IP Managers for Switch Access" on page 6-5.
- You can block all IPv4-based or all IPv6-based management stations from accessing the switch by entering the following commands:
  - To block access to all IPv4 manager addresses while allowing access to IPv6 manager addresses, enter the **ip authorized-managers 0.0.0.0** command.
  - To block access to all IPv6 manager addresses while allowing access to IPv4 manager addresses, enter the **ipv6 authorized-managers ::** command. (The double colon represents an IPv6 address that consists of all zero's: **0:0:0:0:0:0:0:0**.)

- You configure each authorized manager address with Manager or Operator-level privilege to access the switch.
  - Manager privilege allows full access to all web browser and console interface screens for viewing, configuration, and all other operations available in these interfaces.
  - Operator privilege allows read-only access from the web browser and console interfaces.
- When you configure station access to the switch using the Authorized IP Managers feature, the settings take precedence over the access configured with local passwords, TACACS+ servers, RADIUS-assigned settings, port-based (802.1X) authentication, and port security settings.

  As a result, the IPv6 address of a networked management device must be configured with the Authorized IP Managers feature before the switch can authenticate the device using the configured settings from other access security features. If the Authorized IP Managers feature disallows access to the device, then access is denied. Therefore, with authorized IP managers configured, logging in with the correct passwords is not sufficient to access a switch through the network unless the station requesting access is also authorized in the switch's Authorized IP Managers configuration.

Configuring Authorized IP Managers for Switch Access

To configure one or more IPv6-based management stations to access the switch using the Authorized IP Managers feature, enter the **ipv6 authorized-managers** command

Syntax: [no] ipv6 authorized-managers *<ipv6-addr>* [*ipv6-mask*] [access <operator | manager>] access-method [all | ssh | telnet | web | snmp | tftp]

Configures one or more authorized IPv6 addresses to access the switch, where:

ipv6-mask *specifies the mask that is applied to an IPv6 address to determine authorized stations. For more information, see "Using a Mask to Configure Authorized Management Stations" on page 6-5. Default:* **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**.

access <operator | manager> *specifies the level of access privilege granted to authorized stations. Applies only to access through telnet, SSH, and SNMP (version 1,2, and 3). Default:* **Manager.**

access-method [all | ssh | telnet | web | snmp | tftp] *configures access levels by access method and IP address. Each management method can have its own set of authorized managers. Default:* **All**

Using a Mask to Configure Authorized Management Stations

The *ipv6-mask* parameter controls how the switch uses an IPv6 address to determine the IPv6 addresses of authorized manager stations on your network. For example, you can specify a mask that authorizes:

- Single station access
- Multiple station access

Note

Mask configuration is a method for determining the valid IPv6 addresses that are authorized for management access to the switch. In the Authorized IP Managers feature, the mask serves a different purpose than an IPv6 subnet mask and is applied in a different manner.

Configuring Single Station Access

To authorize only one IPv6-based station for access to the switch, enter the IPv6 address of the station and set the mask to **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**.

Notes

If you do not enter a value for the *ipv6-mask* parameter when you configure an authorized IPv6 address, the switch automatically uses **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** as the default mask (see "Configuring Authorized IP Managers for Switch Access" on page 6-5).

If you have ten or fewer management and/or operator stations for which you want to authorize access to the switch, it may be more efficient to configure them by entering each IPv6 address with the default mask in a separate **ipv6 authorized-managers** command.

When used in a mask, "**FFFF**" specifies that each bit in the corresponding 16-bit (hexadecimal) block of an authorized station's IPv6 address must be identical to the same "on" or "off" setting in the IPv6 address entered in the **ipv6 authorized-managers** command. (The binary equivalent of **FFFF** is 1111 1111 1111 1111, where **1** requires the same "on" or "off" setting in an authorized address.)

For example, as shown in Figure 6-1, if you configure a link-local IPv6 address of FE80::202:B3FF:FE1E:8329 with a mask of **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF**, only a station having an IPv6 address of FE80::202:B3FF:FE1E:8329 has management access to the switch.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	The "FFFF" in each hexadecimal block of the mask specifies that only the exact value of each bit in the corresponding block of the IPv6 address is allowed. This mask allows management access only to a station having an IPv6 address of FE80::202:B3FF:FE1E:8329.
IPv6 Address	FE80	0000	0000	0000	202	B3FF	FE1E	8329	

Figure 6-1. Mask for Configuring a Single Authorized IPv6 Manager Station

Configuring Multiple Station Access

To authorize multiple stations to access the switch without having to re-enter the **ipv6 authorized-managers** command for each station, carefully select the IPv6 address of an authorized IPv6 manager and an associated mask to authorize a range of IPv6 addresses.

As shown in Figure 6-2, if a bit in any of the 4-bit binary representations of a hexadecimal value in a mask is "on" (set to 1), then the corresponding bit in the IPv6 address of an authorized station must match the "on" or "off' setting of the same bit in the IPv6 address you enter with the **ipv6 authorized-managers** command.

Conversely, in a mask, a "0" binary bit means that either the "on" or "off" setting of the corresponding IPv6 bit in an authorized address is valid and does not have to match the setting of the same bit in the specified IPv6 address.

Figure 6-2 shows the binary expressions represented by individual hexadecimal values in an *ipv6-mask* parameter.

Hexadecimal Value in an IPv6 Mask	Binary Equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Figure 6-2. Hexadecimal Mask Values and Binary Equivalents

Example. Figure 6-3 shows an example in which a mask that authorizes switch access to four management stations is applied to the IPv6 address: **2001:DB8:0000:0000:244:17FF:FEB6:D37D**. The mask is: **FFFF:FFFF:FFFF:FFF8:FFFF:FFFF:FFFF:FFFC**.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC	The "F" value in the first 124 bits of the mask specifies that only the exact value of each corresponding bit in an authorized IPv6 address is allowed. However, the "C" value in the last four bits of the mask allows four possible combinations (D37C, D37D, D37E, and D37F) in the last block of an authorized IPv6 address.
IPv6 Address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

Figure 6-3. Example: Mask for Configuring Four Authorized IPv6 Manager Stations

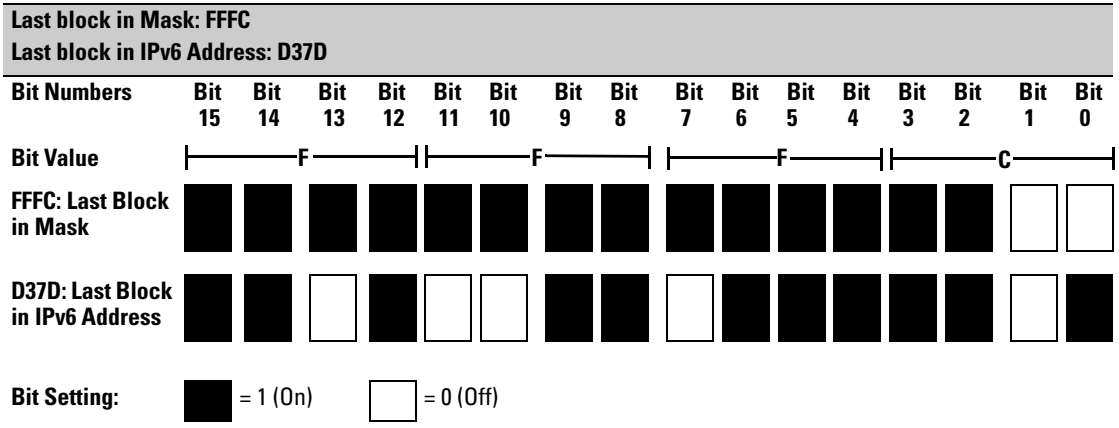

Figure 6-4. Example: How a Mask Determines Four Authorized IPv6 Manager Addresses

As shown in Figure 6-4, if you use a mask of **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC** with an IPv6 address, you can authorize four IPv6-based stations to access the switch. In this mask, all bits except the last two are set to 1 ("on"); the binary equivalent of hexadecimal **C** is 1100.

Therefore, this mask requires the first corresponding 126 bits in an authorized IPv6 address to be the same as in the specified IPv6 address: 2001:DB8:0000:0000:244:17FF:FEB6:D37C. However, the last two bits are set

to 0 (“off”) and allow the corresponding bits in an authorized IPv6 address to be either “on” or “off”. As a result, only the four IPv6 addresses shown in Figure 6-5 are allowed access.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block
IPv6 Mask	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFF	FFFC
IPv6 Address Entered with the “ipv6 authorized-managers” Command	2001	DB8	0000	0000	244	17FF	FEB6	D37D
Other Authorized IPv6 Addresses	2001	DB8	0000	0000	244	17FF	FEB6	D37C
	2001	DB8	0000	0000	244	17FF	FEB6	D37E
	2001	DB8	0000	0000	244	17FF	FEB6	D37F

Figure 6-5. Example: How Hexadecimal C in a Mask Authorizes Four IPv6 Manager Addresses

Example. Figure 6-6 shows an example in which a mask is applied to the IPv6 address: **2001:DB8:0000:0000:244:17FF:FEB6:D37D/64**. The specified mask **FFFF:FFFF:FFFF:FFF8:FFFF:FFFF:FFFF:FFFF** configures eight management stations as authorized IP manager stations.

Note that, in this example, the IPv6 mask is applied as follows:

- Eight management stations in different subnets are authorized by the value of the fourth block (**FFF8**) in the 64-bit prefix ID (**FFFF:FFFF:FFFF:FFF8**) of the mask. (The fourth block of the prefix ID is often used to define subnets in an IPv6 network.)

  The binary equivalent of **FFF8** that is used to specify valid subnet IDs in the IPv6 addresses of authorized stations is: 1111 1111 1111 1000.

  The three “off” bits (1000) in the last part of the this block (**FFF8**) of the mask allow for eight possible authorized IPv6 stations:
  2001:DB8:0000:0000:244:17FF:FEB6:D37D
  2001:DB8:0000:0001:244:17FF:FEB6:D37D
  2001:DB8:0000:0002:244:17FF:FEB6:D37D
  2001:DB8:0000:0003:244:17FF:FEB6:D37D
  2001:DB8:0000:0004:244:17FF:FEB6:D37D
  2001:DB8:0000:0005:244:17FF:FEB6:D37D
  2001:DB8:0000:0006:244:17FF:FEB6:D37D
  2001:DB8:0000:0007:244:17FF:FEB6:D37D

- Each authorized station has the same 64-bit device ID (**244:17FF:FEB6:D37D**) because the value of the last four blocks in the mask is **FFFF** (binary value 1111 1111).

  FFFF requires all bits in each corresponding block of an authorized IPv6 address to have the same "on" or "off" setting as the device ID in the specified IPv6 address. In this case, each bit in the device ID (last four blocks) in an authorized IPv6 address is fixed and can be only one value: 244:17FF:FEB6:D37D.

	1st Block	2nd Block	3rd Block	4th Block	5th Block	6th Block	7th Block	8th Block	Manager- or Operator-Level Access
IPv6 Mask	FFFF	FFFF	FFFF	FFF8	FFFF	FFFF	FFFF	FFFF	In this example, the IPv6 mask allows up to four stations in different subnets to access the switch. This authorized IP manager configuration is useful if only management stations are specified by the authorized IPv6 addresses. Refer to Figure 6-4 for how the bitmap of the IPv6 mask determines authorized IP manager stations.
Authorized IPv6 Address	2001	DB8	0000	0000	244	17FF	FEB6	D37D	

Figure 6-6. Example: Mask for Configuring Authorized IPv6 Manager Stations in Different Subnets

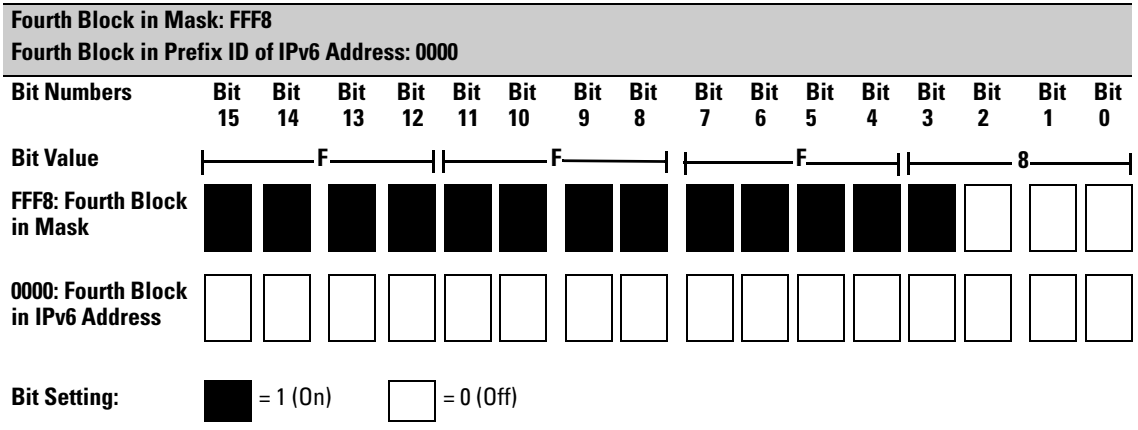

Figure 6-7. Example: How a Mask Determines Authorized IPv6 Manager Addresses by Subnet

Figure 6-7 shows the bits in the fourth block of the mask that determine the valid subnets in which authorized stations with an IPv6 device ID of **244:17FF:FEB6:D37D** reside.

FFF8 in the fourth block of the mask means that bits 3 - 15 of the block are fixed and, in an authorized IPv6 address, must correspond to the "on" and "off" settings shown for the binary equivalent 0000 in the fourth block of the IPv6 address. Conversely, bits 0 - 2 are variable and, in an authorized IPv6 address, may be either "on" (1) or "off" (0).

As a result, assuming that the seventh and eighth bytes (fourth hexadecimal block) of an IPv6 address are used as the subnet ID, only the following binary expressions and hexadecimal subnet IDs are supported in this authorized IPv6 manager configuration:

Authorized Subnet ID in Fourth Hexadecimal Block of IPv6 Address	Binary Equivalent
0000	0000 0000
0001	0000 0001
0002	0000 0010
0003	0000 0011
0004	0000 0100
0005	0000 0101
0006	0000 0110
0007	0000 0111

Figure 6-8. Binary Equivalents of Authorized Subnet IDs (in Hexadecimal)

Displaying an Authorized IP Managers Configuration

Use the **show ipv6 authorized-managers** command to list the IPv6 stations authorized to access the switch; for example:

```
ProCurve# show ipv6 authorized-managers

 IPv6 Authorized Managers
-------------------------------------

 Address : 2001:db8:0:7::5
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
 Access  : Manager

 Address : 2001:db8::a:1c:e3:3
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:fffe
 Access  : Manager

 Address : 2001:db8::214:c2ff:fe4c:e480
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
 Access  : Manager

 Address : 2001:db8::10
 Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00
 Access  : Operator
```

Figure 6-9. Example of "show ipv6 authorized-managers" Output

By analyzing the masks displayed in Figure 6-9, the following IPv6 stations are granted access:

Mask	Authorized IPv6 Addresses	Number of Authorized Addresses
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFC	2001:db8:0:7::4 through 2001:db8:0:7::7	4
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFE	2001:db8::a:1c:e3:2 and 2001:db8::a:1c:e3:3	2
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF	2001:db8::214:c2ff:fe4c:e480	1
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FF00	2001:db8::0 through 2001:db8::FF	256

Figure 6-10. How Masks Determine Authorized IPv6 Manager Addresses

Additional Examples of Authorized IPv6 Managers Configuration

Authorizing Manager Access. The following IPv6 commands authorize manager-level access for one link-local station at a time. Note that when you enter a link-local IPv6 address with the **ipv6 authorized-managers** command, you must also enter a VLAN ID in the format: **%vlan**<*vlan-id*>.

```
ProCurve(config)# ipv6 authorized-managers
fe80::07be:44ff:fec5:c965%vlan2

ProCurve(config)# ipv6 authorized-managers
fe80::070a:294ff:fea4:733d%vlan2

ProCurve(config)# ipv6 authorized-managers
fe80::19af:2cff:fe34:b04a%vlan5
```

If you do not enter an *ipv6-mask* value when you configure an authorized IPv6 address, the switch automatically uses **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** as the default IPv6 mask. Also, if you do not specify an **access** value to grant either Manager- or Operator-level access, by default, the switch assigns Manager access. For example:

```
ProCurve# ipv6 authorized-managers 2001:db8::a8:1c:e3:69
ProCurve# show ipv6 authorized-managers

 IPv6 Authorized Managers
---------------------------

Address : 2001:db8::a8:1c:e3:69
Mask    : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
Access  : Manager
```

If you do not enter a value for *ipv6-mask* in the **ipv6 authorized-managers** command, the default mask of FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF: is applied. The default mask authorizes only the specified station (see "Configuring Single Station Access" on page 6-5).

Figure 6-11. Default IPv6 Mask

The next IPv6 command authorizes operator-level access for sixty-four IPv6 stations: thirty-two stations in the subnets defined by 0x0006 and 0x0007 in the fourth block of an authorized IPv6 address:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8:0000:0007:231:17ff:fec5:c967
ffff:ffff:ffff:fffe:ffff:ffff:ffff:ffe0 access operator
```

The following **ipv6 authorized-managers** command authorizes a single, automatically generated (EUI-64) IPv6 address with manager-level access privilege:

```
ProCurve(config)# ipv6 authorized-managers
::223:04ff:fe03:4501 ::ffff:ffff:ffff:ffff
```

Editing an Existing Authorized IP Manager Entry. To change the mask or access level for an existing authorized IP manager entry, enter the IPv6 address with the new value(s). Any parameters not included in the command are reset to their default values.

The following command replaces the existing mask and access level for IPv6 address 2001:DB8::231:17FF:FEC5:C967 with **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FF00** and **operator**:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8::231:17ff:fec5:c967
ffff:ffff:ffff:ffff:ffff:ffff:ffff:ff00 access operator
```

The following command replaces the existing mask and access level for IPv6 address 2001:DB8::231:17FF:FEC5:3E61 with **FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF** and **manager** (the default values). Note that it is not necessary to enter either of these parameters:

```
ProCurve(config)# ipv6 authorized-managers
2001:db8::a05b:17ff:fec5:3f61
```

Deleting an Authorized IP Manager Entry. Enter only the IPv6 address of the configured authorized IP manager station that you want to delete with the **no** form of the command; for example:

```
ProCurve(config)# no ipv6 authorized-managers
2001:db8::231:17ff:fec5:3e61
```

Secure Shell (SSH) for IPv6

Beginning with software release K.14.01, SSH for IPv4 and IPv6 operate simultaneously with the same command set. Both are enabled in the default configuration, and are controlled together by the same command set.

Secure Shell (SSH) for IPv6 provides the same Telnet-like functions through encrypted, authenticated transactions as SSH for IPv4. SSH for IPv6 provides CLI (console) access and secure file transfer functionality. The following types of transactions are supported:

- Client public-key authentication

  Public keys from SSH clients are stored on the switch. Access to the switch is granted only to a client whose private key matches a stored public key.

- Password-only client authentication

  The switch is SSH-enabled but is not configured with the login method that authenticates a client's public-key. Instead, after the switch authenticates itself to a client, users connected to the client authenticate themselves to the switch by providing a valid password that matches the operator- and/or manager-level password configured and stored locally on the switch or on a RADIUS or TACACS+ server.

- Secure Copy (SCP) and Secure FTP (SFTP) client applications

  You can use either one SCP session or one SFTP session at a given time to perform secure file transfers to and from the switch.

Configuring SSH for IPv6

By default, SSH is automatically enabled for IPv4 and IPv6 connections on a switch. You can use the **ip ssh** command options to reconfigure the default SSH settings to configure the following settings used in SSH authentication for IPv4 and IPv6 connections:

- TCP port number
- timeout period
- file transfer
- MAC type
- cipher type

Syntax:. [no] ip ssh

Enables SSH for on the switch for both IPv4 and IPv6, and activates the connection with a configured SSH server (RADIUS or TACACS+). The **no** *form of the command disables SSH on the switch.*

[cipher < *cipher-type* >]

Specify a cipher type to use for connection.

Valid types are:

- *aes128-cbc*
- *3des-cbc*
- *aes192-cbc*
- *aes256-cbc*
- *rijndael-cbc@lysator.liu.se*
- *aes128-ctr*
- *aes192-ctr*
- *aes256-ctr*

Default: All cipher types are available.
Use the ***no*** *form of the command to disable a cipher type.*

[filetransfer]

Enables SSH on the switch to connect to an SCP or SFTP client application to transfer files to and from the switch over IPv4 or IPv6.

Default: Disabled.

Note: *Enabling filetransfer automatically disables TFTP client and TFTP server functionality.*

For more information, refer to "Secure Copy and Secure FTP for IPv6" on page 6-19.

[mac < *MAC-type* >]

Allows configuration of the set of MACs that can be selected. Valid types are:

- *hmac-md5*
- *hmac-sha1*
- *hmac-sha1-96*
- *hmac-md5-96*

Default: All MAC types are available.
Use the ***no*** *form of the command to disable a MAC type.*

[port < 1-65535 | default >]

TCP port number used for SSH sessions in IPv4 and IPv6 connections (Default: 22).
Valid port numbers are from 1 to 65535, except for port numbers 23, 49, 80, 280,443, 1506, 1513 and 9999, which are reserved for other subsystems.

[public-key < manager | operator > *keystring*

Store a client-generated key for public-key authentication.

manager: *allows manager-level access using SSH public-key authentication.*

operator: *allows operator-level access using SSH public-key authentication.*

keystring:. *a legal SSHv2 (RSA or DSA) public key. The text string for the public key must be a single quoted token. If the keystring contains double-quotes, it can be quoted with single quotes (*'`keystring`'*). The following restrictions for a keystring apply:*

- *A keystring cannot contain both single and double quotes.*
- *A keystring cannot have extra characters, such as a blank space or a new line. (To improve readability, you can add a backlash at the end of each line.)*

For more on configuring and using SSH public keys to authenticate SSH clients connecting to the switch, refer to the chapter titled "Configuring Secure Shell" in the latest Access Security Guide for your switch.

[timeout < 5 - 120 >]

Timeout value allowed to complete an SSH authentication and login on the switch (Default: 120 seconds).

Note For both IPv4 and IPv6, the switch supports only SSH version 2. You cannot set up an SSH session with a client device running SSH version 1.

For more information on how to configure SSH for encrypted, authenticated transactions between the switch and SSH-enabled client devices, refer to the "*Configuring Secure Shell (SSH)*" chapter in the latest *Access Security Guide* for your switch.

Displaying an SSH Configuration

To verify an SSH configuration and display all SSH sessions running on the switch, enter the **show ip ssh** command. Information on all current SSH sessions (IPv4 and IPv6) is displayed.

With SSH running, the switch supports one console session and up to five other SSH and Telnet (IPv4 and IPv6) sessions.

Web browser sessions are also supported, but are not displayed in **show ip ssh** output.

Source IPv6 IP addresses of SSH clients are displayed in hexadecimal format.

```
ProCurve# show ip ssh

  SSH Enabled     : Yes                 Secure Copy Enabled : No
  TCP Port Number : 22                  Timeout (sec)       : 120
  Host Key Type   : RSA                 Host Key Size       : 2048

  Ciphers : aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc,
            rijndael-cbc@lysator.liu.se,aes128-ctr,aes192-ctr,aes256-ctr
  MACs    : hmac-md5,hmac-sha1,hmac-sha1-96,hmac-md5-96

  Ses Type     | Source IP                                        Port
  --- -------- + ------------------------------------------------ -----
  1   console  |
  2   ssh      |10.168.31.114                                     1722
  3   inactive |
  4   inactive |
  5   inactive |
  6   inactive |
```

Displays the current SSH configuration and status.

The switch uses these five SSH settings internally for transactions with clients.

Figure 6-1. Example of an SSH Configuration Display

Secure Copy and Secure FTP for IPv6

You can take advantage of the Secure Copy (SCP) and Secure FTP (SFTP) client applications to provide a secure alternative to TFTP for transferring sensitive switch information, such as configuration files and login information, between the switch and an administrator workstation.

By default, SSH is enabled for IPv4 and IPv6 connections on a switch, and a single command set is used for both IPv4 and IPv6 file transfers.

SCP and SFTP run over an encrypted SSH session, allowing you to use a secure SSH tunnel to:

- Transfer files and update ProCurve software images.
- Distribute new software images with automated scripts that make it easier to upgrade multiple switches simultaneously and securely.

You can perform secure file transfers to and from IPv4 and IPv6 client devices by entering the **ip ssh filetransfer** command.

Syntax: [no] ip ssh filetransfer

Enables SSH on the switch to connect to an SCP or SFTP client application to transfer files to and from the switch.

Use the **no ip ssh filetransfer** *command to disable the switch's ability to perform secure file transfers with an SCP or SFTP client, without disabling SSH on the switch.*

After an IPv6 client running SCP/SFTP successfully authenticates and opens an SSH session on the switch, you can copy files to and from the switch using secure, encrypted file transfers. Refer to the documentation that comes with an SCP or SFTP client application for information on the file transfer commands and software utilities to use.

Notes

Enabling SSH file transfer disables TFTP and Auto-TFTP operation.

The switch supports one SFTP session or one SCP session at a time.

All files on the switch have read-write permission. However, several SFTP commands, such as **create** or **remove**, are not supported and return an error.

For complete information on how to configure SCP or SFTP in an SSH session to copy files to and from the switch, refer to the *"File Transfers"* appendix in the *Management and Configuration Guide* for your switch.

Multicast Listener Discovery (MLD) Snooping

Contents

Overview 7-2

Introduction to MLD Snooping 7-3

Configuring MLD 7-8

Enabling or Disabling MLD Snooping on a VLAN 7-8

Configuring Per-Port MLD Traffic Filters 7-9

Configuring the Querier 7-10

Configuring Fast Leave 7-10

Configuring Forced Fast Leave 7-11

Displaying MLD Status and Configuration 7-12

Current MLD Status 7-12

Current MLD Configuration 7-15

Ports Currently Joined 7-17

Statistics 7-18

Counters 7-20

Overview

Multicast addressing allows one-to-many or many-to-many communication among hosts on a network. Typical applications of multicast communication include audio and video streaming, desktop conferencing, collaborative computing, and similar applications.

Multicast Listener Discovery (MLD) is an IPv6 protocol used on a local link for multicast group management. MLD is enabled per VLAN, and is analogous to the IPv4 IGMP protocol.

MLD snooping is a subset of the MLD protocol that operates at the port level and conserves network bandwidth by reducing the flooding of multicast IPv6 packets.

This chapter describes concepts of MLD snooping and the CLI commands available for configuring it and for viewing its status.

Introduction to MLD Snooping

There are several roles that network devices may play in an IPv6 multicast environment:

- **MLD host**—a network node that uses MLD to "join" (subscribe to) one or more multicast groups
- **multicast router**—a router that routes multicast traffic between subnets
- **querier**—a switch or multicast router that identifies MLD hosts by sending out MLD queries, to which the MLD hosts respond

Curiously enough, a network node that acts as a *source* of IPv6 multicast traffic is only an indirect participant in MLD snooping—it just provides multicast traffic, and MLD doesn't interact with it. (Note, however, that in an application like desktop conferencing a network node may act as both a source and an MLD host; but MLD interacts with that node only in its role as an MLD host.)

A source node creates multicast traffic by sending packets to a multicast address. In IPv6, addresses with the first eight bits set (that is, "FF" as the first two characters of the address) are multicast addresses, and any node that listens to such an address will receive the traffic sent to that address. Application software running on the source and destination systems cooperates to determine what multicast address to use. (Note that this is a function of the application software, not of MLD.)

For example, if several employees engage in a desktop conference across the network, they all need application software on their computers. At the start of the conference, the software on all the computers determines a multicast address of, say, FF3E:30:2001:DB8::101 for the conference. Then any traffic sent to that address can be received by all computers listening on that address.

General operation. Multicast communication can take place without MLD, and by default MLD is disabled. In that case, if a switch receives a packet with a multicast destination address, it floods the packet to all ports in the same VLAN (except the port that it came in on). Any network nodes that are listening to that multicast address will see the packet; all other hosts ignore the packet.

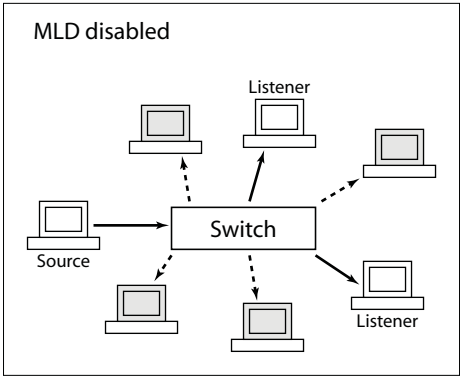

Figure 7-1. Without MLD, multicast traffic is flooded to all ports.

When MLD snooping is enabled on a VLAN, the switch acts to minimize unnecessary multicast traffic. If the switch receives multicast traffic destined for a given multicast address, it forwards that traffic only to ports on the VLAN that have MLD hosts for that address. It drops that traffic for ports on the VLAN that have no MLD hosts (except for a few special cases explained below).

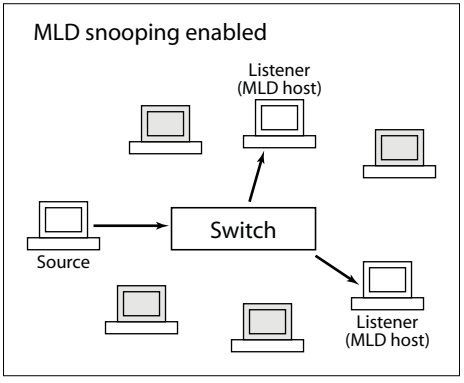

Figure 7-2. With MLD snooping, traffic is sent to MLD hosts.

Note that MLD snooping operates on a single VLAN (though there can be multiple VLANs, each running MLD snooping). Cross-VLAN traffic is handled by a multicast router.

Forwarding in MLD snooping. When MLD snooping is active, a multicast packet is handled by the switch as follows:

- forwarded to ports that have nodes that have joined the packet's multicast address (that is, MLD hosts on that address)
- forwarded toward the querier—If the switch is not the querier, the packet is forwarded out the port that leads to the querier.
- forwarded toward any multicast routers—If there are multicast routers on the VLAN, the packet is forwarded out any port that leads to a router.
- forwarded out administratively forwarded ports—The packet will be forwarded through all ports set administratively to forward mode. (See the description of forwarding modes, below.)
- dropped for all other ports

Each individual port's forwarding behavior can be explicitly set using a CLI command to one of these modes:

- auto (the default mode)—The switch forwards packets through this port based on the MLD rules and the packet's multicast address. In most cases, this means that the switch forwards the packet only if the port connects to a node that is joined to the packet's multicast address (that is, to an MLD host). There is seldom any reason to use a mode other than "auto" in normal operation (though some diagnostics may make use of "forward" or "block" mode).
- forward—The switch forwards all IPv6 multicast packets through the port. This includes IPv6 multicast data and MLD protocol packets.
- block—The switch drops all MLD packets received by the port and blocks all outgoing IPv6 multicast packets through the port, except those packets destined for well known IPv6 multicast addresses. This has the effect of preventing IPv6 multicast traffic from moving through the port.

Note that the switch floods all packets with "well known" IPv6 multicast destination addresses through all ports. Well known addresses are permanent addresses defined by the Internet Assigned Numbers Authority (www.iana.org). IPv6 standards define any address beginning with FF0x/12 (binary 1111 1111 0000) as a well known address.

Listeners and joins. The "snooping" part of MLD snooping arises because a switch must keep track of which ports have network nodes that are MLD hosts for any given multicast address. It does this by keeping track of "joins" on a per-port basis.

A network node establishes itself as an MLD host by issuing a multicast "join" request (also called a multicast "report") for a specific multicast address when it starts an application that listens to multicast traffic. The switch to which the node is connected sees the join request and forwards traffic for that multicast address to the node's port.

Queries. The querier is a multicast router or a switch that periodically asks MLD hosts on the network to verify their multicast join requests. There is one querier for each VLAN, and all switches on the VLAN listen to the responses of MLD hosts to multicast queries, and forward or block multicast traffic accordingly.

All of the ProCurve switches described by this guide have the querier function enabled by default. If there is another device on the VLAN that is already acting as querier, the switch defers to that querier. If there is no device acting as querier, the switch enters an election state and negotiates with other devices on the network (if any) to determine which one will act as the querier.

The querier periodically sends general queries to MLD hosts on each multicast address that is active on the VLAN. The time period that the querier waits between sending general queries is known as the query interval; the MLD standard sets the default query interval to 125 seconds.

Network nodes that wish to remain active as MLD hosts respond to the queries with join requests; in this way they continue to assert their presence as MLD hosts. The switch through which any given MLD host connects to the VLAN sees the join requests and continues forwarding traffic for that multicast address to the MLD host's port.

Leaves. A node acting as an MLD host can be disconnected from a multicast address in two ways:

- It can stop sending join requests to the querier. This might happen if the multicast application quits or the node is removed from the network. If the switch goes for slightly more than two query intervals without seeing a join request from the MLD host, it stops sending multicast traffic for that multicast address to the MLD host's port.
- It can issue a "leave" request. This is done by the application software running on the MLD host. If the MLD host is the only node connected to its switch port, the switch sees the leave request and stops sending multicast packets for that multicast address to that port. (If there is more than one node connected to the port the situation is somewhat more complicated, as explained below under "Fast leaves and forced fast leaves".)

Fast leaves and forced fast leaves. The fast leave and forced fast leave functions can help to prune unnecessary multicast traffic when an MLD host issues a leave request from a multicast address. Fast leave is enabled by default and forced fast leave is disabled by default. Both functions are applied to individual ports.

Which function to use depends on whether a port has more than one node attached to it, as follows:

- If a port has only one node attached to it, then when the switch sees a leave request from that node (an MLD host) it knows that it does not need to send any more multicast traffic for that multicast address to the host's port. If fast leave is enabled (the default setting), the switch stops sending the multicast traffic immediately. If fast leave is disabled, the switch continues to look for join requests from the host in response to group-specific queries sent to the port. The interval during which the switch looks for join requests is brief and depends on the forced fast leave setting: if forced fast leave is enabled for the port, it is equal to the "forced fast leave interval" (typically a couple of seconds or less); if forced fast leave is disabled for the port, the period is about 10 seconds (governed by the MLD standard). When this process has completed the multicast traffic for the group will be stopped (unless the switch sees a new join request).
- If there are multiple nodes attached to a single port, then a leave request from one of those nodes (an MLD host) does not provide enough information for the switch to stop sending multicast traffic to the port. In this situation the fast leave function does not operate. The switch continues to look for join requests from any MLD hosts connected to the port, in response to group-specific queries sent to the port. As in the case described above for a single-node port that is not enabled for fast leave, the interval during which the switch looks for join requests is brief and depends on the forced fast leave setting. If forced fast leave is enabled for the port, it is equal to the "forced fast leave interval" (typically a couple of seconds or less); if forced fast leave is disabled for the port, the period is about 10 seconds (governed by the MLD standard). When this process has completed the multicast traffic for the group will be stopped unless the switch sees a new join request. This reduces the number of multicast packets forwarded unnecessarily.

Configuring MLD

Several CLI commands are available for configuring MLD parameters on a switch.

Enabling or Disabling MLD Snooping on a VLAN

Syntax: [no] ipv6 mld

Note: This command must be issued in a VLAN context.

This command enables MLD snooping on a VLAN. Enabling MLD snooping applies the last-saved or the default MLD configuration, whichever was most recently set.

The **[no]** *form of the command disables MLD snooping on a VLAN.*

MLD snooping is disabled by default.

For example, to enable MLD snooping on VLAN 8:

```
ProCurve# config
ProCurve(config)# vlan 8
ProCurve(vlan-8)# ipv6 mld
```

To disable MLD snooping on VLAN 8:

```
ProCurve(vlan-8)# no ipv6 mld
```

Configuring Per-Port MLD Traffic Filters

Syntax: ipv6 mld [auto <*port-list*> | blocked <*port-list*> | forward <*port-list*>]

Note: *This command must be issued in a VLAN context.*

This command sets per-port traffic filters, which specify how each port should handle MLD traffic. Allowed settings are:

auto—*follows MLD snooping rules: packets are forwarded for joined groups*

blocked—*all multicast packets are dropped, except that packets for well known addresses are forwarded*

forward—*all multicast packets are forwarded*

The default value of the filter is **auto**.

<*port-list*>—*specifies the affected port or range of ports*

For example:

```
ProCurve(vlan-8)# ipv6 mld forward a16-a18
ProCurve(vlan-8)# ipv6 mld blocked a19-a21
ProCurve(vlan-8)# show ipv6 mld vlan 8 config

 MLD Service Vlan Config

  VLAN ID : 8
  VLAN NAME : VLAN8
  MLD Enabled [No] : Yes
  Querier Allowed [Yes] : Yes

  Port Type      | Port Mode Forced Fast Leave Fast Leave
  ---- --------- + --------- ----------------- ----------
  A13  100/1000T | auto      No                Yes
  A14  100/1000T | auto      No                Yes
  A15  100/1000T | auto      No                Yes
  A16  100/1000T | forward   No                Yes
  A17  100/1000T | forward   No                Yes
  A18  100/1000T | forward   No                Yes
  A19  100/1000T | blocked   No                Yes
  A20  100/1000T | blocked   No                Yes
  A21  100/1000T | blocked   No                Yes
  A22  100/1000T | auto      No                Yes
  A23  100/1000T | auto      No                Yes
  A24  100/1000T | auto      No                Yes
```

Figure 7-3. Example of an MLD Configuration with Traffic Filters

Configuring the Querier

Syntax: [no] ipv6 mld querier

> ***Note:*** *This command must be issued in a VLAN context.*
>
> *This command enables the switch to act as querier on a VLAN.*
>
> *The* **[no]** *form of the command disables the switch from acting as querier on a VLAN.*
>
> *The querier function is enabled by default. If another switch or a multicast router is acting as the MLD querier on the VLAN, this switch will defer to that device. If an acting querier stops performing the querier function, all querier-enabled switches and multicast routers on the VLAN will enter an election to determine the next device to act as querier.*

For example, to disable the switch from acting as querier on VLAN 8:

```
ProCurve(vlan-8)# no ipv6 mld querier
```

To enable the switch to act as querier on VLAN 8:

```
ProCurve(vlan-8)# ipv6 mld querier
```

Configuring Fast Leave

Syntax: [no] ipv6 mld fastleave <*port-list*>

> *Note: This command must be issued in a VLAN context.*
>
> *This command enables the fast leave function on the specified ports in a VLAN.*
>
> *The* **[no]** *form of the command disables the fast leave function on the specified ports in a VLAN.*
>
> *The fast leave function is enabled by default.*

For example, to disable fast leave on ports in VLAN 8:

```
ProCurve(vlan-8)# no ipv6 mld fastleave a14-a15
```

To enable fast leave on ports in VLAN 8:

```
ProCurve(vlan-8)# ipv6 mld fastleave a14-a15
```

Configuring Forced Fast Leave

Syntax: [no] ipv6 mld forcedfastleave <*port-list*>

Note: *This command must be issued in a VLAN context.*

This command enables the forced fast leave function on the specified ports in a VLAN.

The **[no]** *form of the command disables the forced fast leave function on the specified ports in a VLAN.*

The forced fast leave function is disabled by default.

For example, to enable forced fast leave on ports in VLAN 8:

```
ProCurve(vlan-8)# ipv6 mld forcedfastleave a19-a20
```

To disable forced fast leave on ports in VLAN 8:

```
ProCurve(vlan-8)# no ipv6 mld forcedfastleave a19-a20
```

Displaying MLD Status and Configuration

Current MLD Status

Syntax: show ipv6 mld

Displays MLD status information for all VLANs on the switch that have MLD configured.

show ipv6 mld vlan <vid>

Displays MLD status for the specified VLAN
vid—*VLAN ID*

For example, a switch with MLD snooping configured on VLANs 8 and 9 might show the following information:

```
ProCurve# show ipv6 mld

  MLD Service Protocol Info

  Total vlans with MLD enabled                : 2
  Current count of multicast groups joined    : 37

  VLAN ID : 8
  VLAN NAME : VLAN8
  Querier Address : fe80::218:71ff:fec4:2f00 [this switch]
  Querier Up Time : 1h:37m:20s
  Querier Expiry Time : 0h:1m:44s

  Ports with multicast routers :


  Active Group Addresses                   Type ExpiryTime Ports
  ---------------------------------------- ---- ---------- --------------------
  ff02::c                                  FILT 0h:4m:9s   A15-A21
  ff02::1:2                                FILT 0h:4m:3s   A21
  ff02::1:3                                FILT 0h:4m:9s   A15-A21
  ff02::1:ff00:42                          FILT 0h:4m:0s   A19
  ff02::1:ff02:2                           FILT 0h:4m:2s   A15
  ff02::1:ff02:3                           FILT 0h:4m:5s   A16
  ff02::1:ff03:2                           FILT 0h:4m:2s   A17
  ff02::1:ff03:3                           FILT 0h:4m:5s   A18
```

Figure 7-4. Example of Displaying the MLD Configuration for All Static VLANs on the Switch

```
 ff02::1:ff04:3                          FILT 0h:4m:5s   A20
 ff02::1:ff05:1                          FILT 0h:4m:3s   A21
 ff02::1:ff0b:2dfe                       FILT 0h:3m:59s  A17
 ff02::1:ff0b:d7d9                       FILT 0h:4m:4s   A15
 ff02::1:ff0b:da09                       FILT 0h:4m:5s   A18
 ff02::1:ff0b:dc38                       FILT 0h:4m:3s   A19
 ff02::1:ff0b:dc8d                       FILT 0h:4m:4s   A20
 ff02::1:ff0b:dd56                       FILT 0h:4m:0s   A16
 ff02::1:ff12:e0cd                       FILT 0h:4m:5s   A21
 ff02::1:ff4e:98a5                       FILT 0h:4m:0s   A17
 ff02::1:ff57:21a1                       FILT 0h:3m:58s  A20
 ff02::1:ff6b:dd51                       FILT 0h:4m:0s   A15
 ff02::1:ff7b:ac55                       FILT 0h:4m:5s   A16
 ff02::1:ff8f:61ea                       FILT 0h:4m:1s   A19
 ff02::1:ffc8:397b                       FILT 0h:4m:0s   A18
ff3e:30:2001:db8:8:0:7:101               FILT 0h:4m:4s   A15,A18,A21
 ff3e:30:2001:db8:8:0:7:102              FILT 0h:4m:13s  A16,A19

 VLAN ID : 9
 VLAN NAME : VLAN9
 Querier Address : fe80::218:71ff:fec4:2f00 [this switch]
 Querier Up Time : 1h:37m:22s
 Querier Expiry Time : 0h:1m:43s

Ports with multicast routers :

 Active Group Addresses                  Type ExpiryTime Ports
 --------------------------------------- ---- ---------- --------------------
 ff02::c                                 FILT 0h:4m:12s  B3,B5,B7
 ff02::1:3                               FILT 0h:4m:12s  B3,B5,B7
 ff02::1:ff02:4                          FILT 0h:4m:4s   B3
 ff02::1:ff03:4                          FILT 0h:3m:59s  B5
 ff02::1:ff04:4                          FILT 0h:4m:12s  B7
 ff02::1:ff0b:dc64                       FILT 0h:4m:0s   B7
 ff02::1:ff0b:dcf3                       FILT 0h:4m:2s   B3
 ff02::1:ff0b:dd5c                       FILT 0h:4m:4s   B5
 ff02::1:ff34:a69e                       FILT 0h:4m:1s   B5
 ff02::1:ff8e:11d5                       FILT 0h:3m:57s  B7
 ff02::1:ffea:2c4f                       FILT 0h:3m:58s  B3
```

Figure 7-5. Continuation of Figure 7-4

The following information is shown for each VLAN that has MLD snooping enabled:

- VLAN ID number and name
- Querier address: IPv6 address of the device acting as querier for the VLAN
- Querier up time: the length of time in seconds that the querier has been acting as querier
- Querier expiry time: If this switch is the querier, this is the amount of time until the switch sends the next general query. If this switch is not the querier, this is the amount of time in seconds until the current querier is considered inactive (after which a new querier election is held).
- Ports with multicast routers: ports on the VLAN that lead toward multicast routers (if any)
- Multicast group address information for each active group on the VLAN, including:
  - the multicast group address
  - the type of tracking for multicast joins: standard or filtered. If MLD snooping is enabled, port-level tracking results in filtered groups. If MLD snooping is not enabled, joins result in standard groups being tracked by this device. In addition, if hardware resources for multicast filtering are exhausted, new joins may result in standard groups even though MLD snooping is enabled.
  - expiry time: the time until the group expires if no joins are seen
  - the ports that have joined the multicast group

The group addresses you see listed typically result from several network functions. In our example, several of the addresses at the top of the list for each VLAN are IANA well known addresses (see www.iana.org/assignments/ipv6-multicast-addresses); the addresses in the form of ff02::1:ffxx:xxxx are solicited-node multicast addresses (used in IPv6 Neighbor Discovery); and the addresses beginning with ff3e are group addresses used by listeners to streaming video feeds.

Current MLD Configuration

Syntax: show ipv6 mld config

Displays current global MLD configuration for all MLD-enabled VLANS on the switch.

show ipv6 vlan <*vid*> config

Displays current MLD configuration for the specified VLAN, including per-port configuration information.
vid—*VLAN ID*

For example, the general form of the command might look like this:

```
ProCurve# show ipv6 mld config

 MLD Service Config

  Control unknown multicast  [Yes] : Yes
  Forced fast leave timeout  [4] : 4

  VLAN ID VLAN NAME       MLD Enabled Querier Allowed
  ------- --------------- ----------- ---------------
  8       VLAN8           Yes         Yes
  9       VLAN9           Yes         Yes
```

Figure 7-6. Example of a Global MLD Configuration

The following information, for all MLD-enabled VLANs, is shown:

- Control unknown multicast: If this is set to YES, any IPv6 multicast packets that are not joined by an MLD host will be sent only to ports that have detected a multicast router or ports that are administratively forwarded. If this is set to NO (or if MLD snooping is disabled), unjoined IPv6 multicast packets will be flooded out all ports in the VLAN.
- Forced fast leave timeout: the interval between an address specific query and a forced fast leave (assuming no response), in tenths of seconds
- For each VLAN that has MLD enabled:
  - VLAN ID and name
  - whether MLD is enabled on the VLAN (default NO, but the VLAN will not show up on this list unless MLD is enabled)
  - whether the switch can act as querier for the VLAN (default YES)

The specific form of the command might look like this:

```
ProCurve# show ipv6 mld vlan 8 config

 MLD Service Vlan Config

  VLAN ID : 8
  VLAN NAME : VLAN8
  MLD Enabled [No] : Yes
  Querier Allowed [Yes] : Yes

  Port Type      | Port Mode Forced Fast Leave Fast Leave
  ---- --------- + --------- ----------------- ----------
  A13  100/1000T | auto      No                Yes
  A14  100/1000T | auto      No                Yes
  A15  100/1000T | auto      No                Yes
  A16  100/1000T | auto      No                Yes
  A17  100/1000T | auto      No                Yes
  A18  100/1000T | auto      No                Yes
  A19  100/1000T | auto      No                Yes
  A20  100/1000T | auto      No                Yes
  A21  100/1000T | auto      No                Yes
  A22  100/1000T | auto      No                Yes
  A23  100/1000T | auto      No                Yes
  A24  100/1000T | auto      No                Yes
```

Figure 7-7. Example of an MLD Configuration for a Specific VLAN

The following information is shown, if the specified VLAN is MLD-enabled:

- VLAN ID and name
- whether MLD is enabled on the VLAN (default NO, but the information for this VLAN will be listed only if MLD is enabled)
- whether the switch is allowed to act as querier on the VLAN

Ports Currently Joined

Syntax: show ipv6 mld vlan *<vid>* group

Lists the ports currently joined for all IPv6 multicast group addresses in the specified VLAN
vid—*VLAN ID*

show ipv6 mld vlan *<vid>* group *<ipv6-addr>*

Lists the ports currently joined for the specified IPv6 multicast group address in the specified VLAN
vid—*VLAN ID*
ipv6-addr—*address of the IPv6 multicast group for which you want information*

For example, the general form of the command is shown below. The specific form the the command is similar, except that it lists the port information for only the specified group.

```
ProCurve# show ipv6 mld vlan 9 group

 MLD Service Protocol Group Info

 VLAN ID : 9
 VLAN Name : VLAN9

  Filtered Group Address : ff02::c
 Last Reporter : fe80::7061:4b38:dbea:2c4f
  ExpiryTime : 0h:2m:19s

  Port Port Type | Port Mode ExpiryTime
  ---- --------- + --------- --------------------
  B3   100/1000T | auto      0h:2m:19s
  B5   100/1000T | auto      0h:2m:18s

    .
    .
    .

 Filtered Group Address : ff3e:30:2001:db8:9:0:7:111
  Last Reporter : fe80::7061:4b38:dbea:2c4f
  ExpiryTime : 0h:4m:14s

  Port Port Type | Port Mode ExpiryTime
  ---- --------- + --------- --------------------
  B3   100/1000T | auto      0h:4m:14s
  B5   100/1000T | auto      0h:4m:09s
```

Figure 7-8. Example of Ports Joined to Multicast Groups in a Specific VLAN

The following information is shown:

- VLAN ID and name
- port information for each IPv6 multicast group address in the VLAN (general group command) or for the specified IPv6 multicast group address (specific group command):
  - group multicast address
  - last reporter: last MLD host to send a join to the group address
  - group expiry time: the time until the group expires if no further joins are seen
  - port name for each port
  - port type for each port: Ethernet connection type
  - port mode for each port: auto (follows MLD snooping rules; that is, packets are forwarded for joined groups), forward (all multicast packets are forwarded to this group), or blocked (all multicast packets are dropped, except that packets for well-known addresses are forwarded)
  - expiry time for each port: amount of time until this port is aged out of the multicast address group, unless a join is received

Statistics

Syntax: show ipv6 mld statistics

Shows MLD statistics for all MLD-enabled VLANs

Syntax: show ipv6 mld vlan <vid> statistics

Shows MLD statistics for the specified VLAN
vid—*VLAN ID*

The general form the of the command shows the total number of MLD-enabled VLANs and a count of multicast groups currently joined. Both forms of the command show VLAN IDs and names, as well as the number of filtered and standard multicast groups and the total number of multicast groups.

For example, the general form of the command:

```
ProCurve# show ipv6 mld statistics

 MLD Service Statistics

  Total vlans with MLD enabled               : 2
  Current count of multicast groups joined   : 36


 MLD Joined Groups Statistics

  VLAN ID VLAN NAME    filtered     standard     total
  ------- ------------ ------------ ------------ ------------
  8       VLAN8        26           0            26
  9       VLAN9        10           0            10
```

Figure 7-9. Example of MLD Statistics for All VLANs Configured

And the specific form of the command:

```
ProCurve# show ipv6 mld vlan 8 statistics

 MLD Statistics

  VLAN ID : 8
  VLAN NAME : VLAN8

  Number of Filtered Groups     : 26
  Number of Standard Groups     : 0
  Total Multicast Groups Joined : 26
```

Figure 7-10. Example of MLD Statistics for a Single VLAN

Counters

Syntax: show ipv6 mld vlan *<vid>* counters

Displays MLD counters for the specified VLAN
vid—*VLAN ID*

```
ProCurve# show ipv6 mld vlan 8 counters

 MLD Service Vlan Counters

  VLAN ID : 8
  VLAN NAME : VLAN8

     General Query Rx                   : 2
     General Query Tx                   : 0
     Group Specific Query Rx            : 0
     Group Specific Query Tx            : 0
     V1 Member Report Rx                : 1589
     V2 Member Report Rx                : 15
     Leave Rx                           : 30
     Unknown MLD Type Rx                : 0
     Unknown Pkt Rx                     : 0
     Forward to Routers Tx Counter      : 83
     Forward to Vlan Tx Counter         : 48
     Port Fast Leave Counter            : 4
     Port Forced Fast Leave Counter     : 0
     Port Membership Timeout Counter   : 28
```

Figure 7-11. Example of MLD Counters for a Single VLAN

The following information is shown:

- VLAN number and name
- For each VLAN:
  - number of general queries received
  - number of general queries sent
  - number of group-specific queries received
  - number of group-specific queries sent
  - number of MLD version 1 member reports (joins) received
  - number of MLD version 2 member reports (joins) received
  - number of leaves received
  - number of MLD packets of unknown type received
  - number of packets of unknown type received
  - number of packets forwarded to routers on this VLAN
  - number of times a packet has been forwarded to all ports on this VLAN
  - number of fast leaves that have occurred
  - number of forced fast leaves that have occurred
  - number of times a join has timed out on this VLAN

8

IPv6 Access Control Lists (ACLs)

Contents

Introduction 8-4

Overview of Options for Applying IPv6 ACLs on the Switch 8-6

Static ACLS 8-6

RADIUS-Assigned ACLs 8-6

Command Summary for Configuring ACLs 8-7

Command Summary for Enabling, Disabling, and Displaying ACLs . 8-8

Terminology 8-9

Overview 8-13

Types of IPv6 ACLs 8-13

Concurrent IPv4 and IPv6 ACLs 8-13

IPv6 ACL Applications 8-13

VACL Applications 8-15

IPv6 Static Port ACL Applications 8-16

RADIUS-Assigned (Dynamic) Port ACL Applications 8-16

Multiple ACL Assignments on an Interface 8-18

Features Common to All ACL Applications 8-21

General Steps for Planning and Configuring ACLs 8-22

IPv6 ACL Operation 8-24

Introduction 8-24

The Packet-filtering Process 8-24

Planning an ACL Application 8-28

IPv6 Traffic Management and Improved Network Performance . . . 8-28

Security 8-29

Guidelines for Planning the Structure of an ACL 8-30

ACL Configuration and Operating Rules 8-31

How an ACE Uses a Prefix To Screen Packets for SA and DA Matches 8-33
Prefix Usage Differences Between ACLs and Other IPv6 Addressing 8-34
Configuring and Assigning an IPv6 ACL 8-35
General Steps for Implementing IPv6 ACLs 8-35
Permit/Deny Options 8-36
ACL Configuration 8-36
ACL Configuration Structure 8-38
ACL Configuration Factors 8-40
The Sequence of Entries in an ACL Is Significant 8-40
Allowing for the Implied Deny Function 8-41
A Configured ACL Has No Effect Until You Apply It to an Interface 8-42
You Can Assign an ACL Name to an Interface Even if the ACL Has Not Been Configured 8-42
Using the CLI To Create an ACL 8-42
General ACE Rules 8-43
Using CIDR Notation To Enter the IPv6 ACL Prefix Length . . . 8-43
Configuration Commands 8-45
Command Summary for Configuring ACLs 8-45
Command Summary for Enabling, Disabling, and Displaying ACLs 8-46
Overview 8-46
Commands To Create, Enter, and Configure an ACL 8-47
Adding or Removing an ACL Assignment On an Interface 8-62
Filtering Switched IPv6 Traffic Inbound on a VLAN 8-62
Filtering Inbound IPv6 Traffic Per Port and Trunk 8-63
Deleting an ACL 8-65
Editing an Existing ACL 8-66
General Editing Rules 8-66
Sequence Numbering in ACLs 8-66
Inserting an ACE in an Existing ACL 8-68
Deleting an ACE from an Existing ACL 8-70
Resequencing the ACEs in an IPv6 ACL 8-71
Attaching a Remark to an ACE 8-73

Operating Notes for Remarks 8-76

Displaying ACL Configuration Data 8-78

Display an ACL Summary 8-79

Display the Content of All ACLs on the Switch 8-80

Display the IPv4 and IPv6 VACL Assignments for a VLAN 8-81

Display Static Port (and Trunk) ACL Assignments 8-82

Displaying the Content of a Specific ACL 8-83

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File 8-86

Creating or Editing ACLs Offline 8-87

Creating or Editing an ACL Offline 8-87

The Offline Process 8-87

Example of Using the Offline Process 8-88

Testing and Troubleshooting ACLs 8-92

Enable IPv6 ACL “Deny” Logging 8-92

Requirements for Using IPv6 ACL Logging 8-92

ACL Logging Operation 8-93

Enabling ACL Logging on the Switch 8-93

Monitoring Static ACL Performance 8-96

Example of ACL Performance Monitoring 8-98

IPv6 Counter Operation with Multiple Interface Assignments . 8-99

IPv4 Counter Operation with Multiple Interface Assignments 8-101

General ACL Operating Notes 8-105

Introduction

An Access Control List (ACL) contains one or more Access Control Entries (ACEs) specifying the criteria the switch uses to either permit (forward) or deny (drop) IP packets traversing the switch's interfaces. This chapter describes how to configure, apply, and edit static IPv6 ACLs for filtering IPv6 traffic in a network populated with the switches covered by this guide, and how to monitor IPv6 ACL actions.

Note

Because the switches covered by this guide operate in an IPv4/IPv6 dual stack mode, IPv6 and IPv4 ACLs can operate simultaneously in these switches. However:

- Static IPv6 ACLs and IPv4 ACLs do not filter each other's traffic.
- IPv6 and IPv4 ACEs cannot be configured in the same static ACL.
- RADIUS-assigned ACLs can be configured to filter either IPv4 traffic only, or both IPv4 and IPv6 traffic. Refer to "RADIUS-Assigned ACLs" on page 8-6.

In this chapter, unless otherwise noted:

- The term "ACL" refers to IPv6 ACLs.
- Descriptions of ACL operation apply only to IPv6 traffic.

For information on configuring and applying static IPv4 ACLs, refer to the chapter titled "IPv4 Access Control Lists (ACLs)" in the *Access Security Guid*e for your switch.

.

Feature	Default	CLI
Configure IPv6 ACLs	None	8-35
Enable or Disable an ACL	n/a	8-62
Display ACL Configuration Data	n/a	8-78
Delete an ACL	n/a	8-65
Editing an Existing ACL	n/a	8-66
Creating or Editing an ACL Offline Using TFTP	n/a	8-87
Enable ACL Logging	n/a	8-93

IPv6 traffic filtering with ACLs can help to improve network performance and restrict network use by creating policies for:

- **Switch Management Access:** Permits or denies in-band management access. This includes limiting and/or preventing the use of designated protocols that run on top of IPv6, such as TCP, UDP, ICMP, and others. Also included are the use of DSCP criteria, and control for application transactions based on source and destination IPv6 addresses and transport layer port numbers.
- **Application Access Security:** Eliminates unwanted IPv6 traffic in a path by filtering IPv6 packets where they enter or leave the switch on specific VLAN interfaces.

The ACLs described in this chapter can filter IPv6 traffic to or from a host, a group of contiguous hosts, or entire subnets.

Caution

The ACLs described in this chapter can enhance network security by blocking selected IPv6 traffic, and can serve as part of your network security program. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IPv6 packet transmissions, they should not be relied upon for a complete security solution.*

Static IPv6 ACLs on the switches covered by this manual do not screen non-IPv6 traffic such as IPv4, AppleTalk, and IPX packets.

Overview of Options for Applying IPv6 ACLs on the Switch

To apply IPv6 ACL filtering, assign a configured IPv6 ACL to the interface on which you want the traffic filtering to occur. VLAN IPv6 traffic ACLs can be applied statically using the switch configuration. Port traffic ACLs can be applied either statically or dynamically (using a RADIUS server).

Static ACLS

Static ACLs are configured on the switch. To apply a static ACL, you must assign it to an interface (VLAN or port). The switch supports two static ACL applications:

- **VLAN ACL (VACL):** A VACL is an ACL configured on a VLAN to filter IPv6 traffic entering the switch on that VLAN interface and having a destination on the same VLAN.
- **Static Port ACL:** A static port ACL is an ACL configured on a port to filter IPv6 traffic entering the switch on that port.

RADIUS-Assigned ACLs

A RADIUS-assigned ACL for filtering traffic from a specific client or group of clients is configured on a RADIUS server. When the server authenticates a client associated with that ACL, the ACL is assigned to filter the inbound IP traffic received from the authenticated client through the port on which the client is connected to the switch. If the RADIUS server supports both IPv4 and IPv6 ACEs, then the ACL assigned by the server can be configured to filter both traffic types, or just the IPv4 traffic. When the client session ends, the ACL is removed from the port. The switch allows as many RADIUS-assigned ACLs on a port as it allows authenticated clients. For information on RADIUS-assigned ACLs, refer to the chapter titled, “Configuring RADIUS Server Support for Switch Services” in the latest *Access Security Guide* for your switch.

Note

This chapter describes the IPv6 ACL applications you can statically configure on the switch. For information on static IPv4 ACL applications, refer to the chapter titled “IPv4 Access Control Lists (ACLs)” in the latest *Access Security Guide* for your switch.

Command Summary for Configuring ACLs

Create an IPv6 ACL *or* Add an ACE to the End of an Existing IPv6 ACL	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# < deny | permit > < ipv6 | esp | ah | sctp | *ipv6-protocol-nbr* > < any | host <*SA* > | *SA*/< *prefix-length* >> < any | host < *DA* > | *DA*/< *prefix-length* >> < tcp | udp > < any | host <*SA* > | *SA*/< *prefix-length* > > [*comparison-operator* < *value* >] < any | host < *DA* > | *DA*/< *prefix-length* >> [*comparison-operator* < *value* >] [established][1] [ack] [fin] [rst] [syn][2] < icmp > < any | host < *SA* > | *SA* /< *prefix-length* >> < any | host < *DA* > | *DA* /< *prefix-length* >> [0 - 255 [0 - 255] | *icmp-message*] [dscp < *precedence* | *codepoint* >] [log][3]	8-47
Insert an ACE or a remark by Assigning a Sequence Number	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# < *seq-#* > < deny | permit | remark> *The deny and permit keywords use the options shown above for "Create an IPv6 ACL".*	8-68
Delete an ACE or a Remark (or both) by Sequence Number	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# no < *seq-#* > [remark] (**Note:** You can also delete an ACE by entering no < permit | deny > followed by the settings explicitly configured for that ACE.)	8-70
Resequence the ACEs in an ACL	ProCurve(config)# ipv6 access-list resequence < *name-str* > < *starting-#* > < *increment* >.	8-71

[1]TCP only.
[2]TCP flag (control bit) options for destination TCP.
[3]The **log** function is available only for "deny" ACLs, and generates a message only when there is a "deny" match.

— *Continued* —

— Continued from preceding page. —

Action	Command(s)	Page
Enter a Remark	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# remark < *remark-str* >	8-73 8-75
Remove a Remark: – Immediately After Entry – After entry of an ACE	 ProCurve(config-ipv6-acl)# no remark ProCurve(config-ipv6-acl)#no < *seq-#* > remark	
Delete an IPv6 ACL	ProCurve(config)# no ipv6 access-list < *name-str* > vlan	8-65

Command Summary for Enabling, Disabling, and Displaying ACLs

Enable or Disable an IPv6 VACL	ProCurve(config)# [no] vlan < *vid* > ipv6 access-group < *name-str* > vlan	
Enable or Disable a Static Port ACL	ProCurve(config)# [no] interface < *port-list* | trk*x* > ipv6 access-group < *name-str* > in ProCurve(eth-< *port-list* >| trk*x* >)# [no] ipv6 access-group < *name-str* > in	
Displaying ACL Data	ProCurve(config)# show access-list ProCurve(config)# show access-list < *acl-name-str* > [config] ProCurve(config)# show access-list config ProCurve(config)# show access-list ports < *port-list* > ProCurve(config)# show access-list vlan < *vid* > ProCurve(config)# show access-list radius < *port-list* | all > ProCurve(config)# show access-list resources	8-78

Terminology

Access Control Entry (ACE): A policy consisting of criteria and an action (permit or deny) to execute on a packet if it meets the criteria. For IPv6 ACEs, the elements composing the criteria include:

- source IPv6 address and prefix length
- destination IPv6 address and prefix length
- either of the following:
  - all IPv6 traffic
  - IPv6 traffic of a specific IPv6 protocol (For TCP, UDP, and ICMP, the criteria can include either a specific sub-type within the protocol or all traffic of the protocol type.)
- option to log packet matches with **deny** ACEs
- optional use of DSCP (precedence and ToS settings)

Access Control List (ACL): A list (or set) consisting of one or more explicitly configured Access Control Entries (ACEs) and terminating with an implicit **deny ipv6 any any** ACE. Each ACE in an IPv6 ACL includes layer-3 IPv6 source and destination criteria and IPv6 protocol-specific criteria. IPv6 ACLs can be applied in any of the following ways:

- VACL: an ACL assigned to filter inbound IPv6 traffic on a specific VLAN configured on the switch
- Static Port ACL: an ACL assigned to filter inbound IPv6 traffic on a specific switch port
- RADIUS-Assigned ACL: dynamic ACL assigned to a port by a RADIUS server to filter inbound IPv4 and IPv6 traffic from an authenticated client on that port (Refer to the chapter titled “Configuring RADIUS Server Support for Switch Services” in the latest *Access Security Guide* for your switch.)

Static ACLs are configured in switch memory with an alphanumeric name, and can be assigned to a VLAN as a VACL, and to a port list (or static trunk). (RADIUS-assigned ACLs are configured on a RADIUS server, and are identified by the associated client credentials instead of an alphanumeric name.)

ACE: See “Access Control Entry”.

ACL: See “Access Control List”.

ACL ID: An alphanumeric string used to identify an ACL. See also ***identifier*** and ***name-str***.
Note: RADIUS-assigned ACLs are identified by client authentication data and do not use the ACL ID strings described in this chapter.

ACL Prefix: Follows any IPv6 address listed in an IPv6 ACE. Analogous to the ACL mask used with IPv4 ACEs. Specifies the number of leftmost, contiguous bits in a packet's corresponding IPv6 addressing that must exactly match the IPv6 addressing in the ACE, and which bits need not match (wildcards). Refer to "How an ACE Uses a Prefix To Screen Packets for SA and DA Matches" on page 8-33.)

Address Family: Used in this manual to refer to the version of the IP protocol running on the switch; IPv4 and IPv6.

CIDR: The acronym for Classless Inter-Domain Routing. In IPv6 ACEs, CIDR notation is used to specify the prefix length for SA and DA address criteria. For example, the length of the following prefix includes the first 48 bits of an address: `2001:db8:101::/48`

DA: The acronym for *Destination Address*. In an IPv6 packet, this is the destination IPv6 address carried in the header, and identifies the packet's destination. This is the second of two IPv6 addresses used in an ACE to determine whether there is a match between an IPv6 packet and the ACE. See also "SA".

Deny: An ACE configured with this action causes the switch to drop an IPv6 packet for which there is a match within an applicable ACL.

Empty ACL: An ACL that is not populated with any explicit ACEs, and functions only as a placeholder. An ACL exists in this state if any one of the following occurs:

- An ACL identifier has been created in the running config file with the **ipv6 access-list < *name-str* >** command, but no explicit ACEs exist in the ACL.
- An ACL identifier has been assigned to an interface without first populating the ACL with ACEs. If the empty ACL did not already exist in the running config file, assigning the identifier to an interface automatically creates the empty ACL in the running config file.
- An ACL configured with one or more explicit ACEs has been deleted from the running config file while the ACL is still assigned to an interface.

Note that an empty ACL does not include an Implicit Deny and does not filter traffic. However, if you configure any ACE in an empty ACL that is already assigned to an interface, the ACL immediately begins filtering traffic, which includes application of the Implicit Deny.

identifier*:** A term used in ACL syntax statements to represent the alphanumeric name by which the ACL can be accessed. An identifier can have up to 64 characters. See also ***NAME-STR.
Note: RADIUS-assigned ACLs are identified by client authentication criteria and do not use the identifiers described in this chapter.

Implicit Deny: If the switch finds no matches between an IPv6 packet and the configured criteria in an applicable ACL, then the switch denies (drops) the packet with an implicit **deny ipv6 any any** function. You can pre-empt the Implicit Deny in a given ACL by configuring a **permit ipv6 any any** as the last explicit ACE in the ACL. Doing so permits any packet that is not explicitly permitted or denied by other ACEs configured sequentially earlier in the ACL.
Note: Beginning with software release K.14.01, any dynamically created ACL will include an implicit deny for both Ipv4 and IPv6 traffic, regardless of the address family capabilities of the server. Refer to "RADIUS-Assigned ACLs" on page 8-6.

Inbound Traffic: For the purpose of defining where the switch applies IPv6 ACLs to filter traffic, *inbound traffic* is a packet that meets one of the following criteria:

- VLAN ACL (VACL): Inbound traffic is a packet entering the switch on a VLAN interface (or a subnet in a multinetted VLAN).
- Static Port ACL: Inbound traffic is a packet entering the switch on the port.
- RADIUS-Assigned ACL: Where a RADIUS server has authenticated a client and assigned an ACL to the port to filter the client's IPv6 traffic, inbound traffic is a packet entering the switch from that client. (Note that IPv4 traffic-filtering is automatically included in a RADIUS-assigned ACL configured to filter IPv6 traffic.)

NAME-STR*:** The term used in ACL syntax statements to represent the "name string"; the alphanumeric string used to identify the ACL. A name string allows up to 64 alphanumeric characters. See also ***IDENTIFIER and **ACL-ID**.

Outbound Traffic: For defining the points where the switch applies an RACL (Routed ACL) to filter traffic, outbound traffic is routed traffic *leaving the switch* through a VLAN interface (or a subnet in a multinetted VLAN). "Outbound traffic" can also apply to switched traffic leaving the switch

on a VLAN interface, but outbound, switched traffic is not filtered by ACLs. In software release K.14.01, RACLs are supported for IPv4 traffic, but not for IPv6 traffic. (Refer also to "IPv6 ACL Applications" on page 8-13.)

Permit: An ACE configured with this action allows the switch to forward an IPv6 packet for which there is a match.

Permit Any Forwarding: An ACE configured with this action causes the switch to forward IPv6 packets that have not been permitted or denied by earlier ACEs in the list.

Prefix Length: In an IPv6 ACE, a network prefix is used to specify the leftmost contiguous bits in a packet's SA and DA that must match the bit settings defined in the SA and DA configured in the ACE. The prefix length is specified (in CIDR format) by ***/nn*** immediately following the specified SA or DA address. For example, if the SA prefix in an ACE is 2001:db8:127::/48, then the first 48 bits in the SA of a packet being compared to that ACE must be the same to allow a match. In this case, bits 49 through 128 are not compared and are termed a "wildcard". See also **Wildcard** on page 8-13.

RADIUS-Assigned ACL: An ACL assigned by a RADIUS server to a port to filter inbound IP traffic from a client authenticated by the server for that port.

A RADIUS-assigned ACL can be configured (on a RADIUS server) to filter inbound IPv4 and IPv6 traffic, or just IPv4 traffic. When the client session ends, the RADIUS-assigned ACL for that client is removed from the port. See also "Implicit Deny".

***remark-str*:** The term used in ACL syntax statements to represent the variable "remark string"; a set of alphanumeric characters you can include as a remark in an ACL. A remark string allows up to 100 characters and must be delimited by single or double quotes if any spaces are included in the string.

SA: The acronym for *Source Address*. In an IPv6 packet, this is the source IPv6 address carried in the header, and identifies the packet's sender. This is the first of two IPv6 addresses used in an ACE to determine whether there is a match between a packet and the ACE. See also "DA".

***seq-#*:** The term used in ACL syntax statements to represent the sequence number variable used to insert an ACE within an existing list. The range allowed for sequence numbers is 1 - 2147483647.

Static Port ACL: An ACL statically configured on a specific port, group of ports, or trunk. A static port ACL filters incoming IPv6 traffic on the port.

VACL: See "VLAN ACL".

VLAN ACL (VACL): An ACL applied to all IPv6 traffic entering the switch on a given VLAN interface. See also "Access Control List".

Wildcard: The bits in an SA or DA of a packet that are ignored when determining whether the packet is a match for a given ACE. That is, when the switch is comparing the address bits in a packet header with the address bits specified in a given IPv6 ACE, only the address bits included in the prefix length in the ACE are significant. The remaining bits—those to the right of the bits specified by the prefix length—comprise a wildcard and can be either on or off. See also **Prefix Length** on page 8-12.

Overview

Types of IPv6 ACLs

A permit or deny policy for IPv6 traffic you want to filter is based on source and destination IPv6 address, plus other IPv6 protocol factors such as TCP/UDP, ICMP, and DSCP.

Concurrent IPv4 and IPv6 ACLs

The switches covered by this guide support concurrent configuration and operation of IPv4 and IPv6 ACLs. For information on IPv4 ACLs, refer to the *Access Security Guide* for your switch.

IPv6 ACL Applications

ACL filtering is applied to IPv6 traffic as follows:

- VLAN ACL (VACL): On a VLAN configured with a VACL, filters inbound IPv6 traffic. On a multinetted VLAN, this includes inbound IPv6 traffic from any subnet.
- Static port ACL: Filters inbound IPv6 traffic on the port.

- RADIUS-assigned ACL: on a port having an ACL assigned by a RADIUS server to filter an authenticated client's traffic, filters inbound IPv4 and IPv6 traffic (or IPv4-only traffic) from that client

  (For information on RADIUS-assigned ACLs, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the latest *Access Security Guide* for your switch.)

VACL Applications

IPv6 VACLs filter traffic entering the switch on a VLAN configured with the "VLAN" ACL option.

vlan < *vid* > ipv6 access-group < *identifier* > vlan

For example, in figure 8-1:

- Assigning an IPv6 VACL to VLAN 1 filters inbound IPv6 traffic received from clients on the 2001:db8:0:111:: network.
- Assigning an IPv6 VACL to VLAN 2 (which is subnetted) filters inbound IPv6 traffic from clients on the 2001:db8:0:22a::, 2001:db8:0:22b::, and 2001:db8:0:22c:: networks.

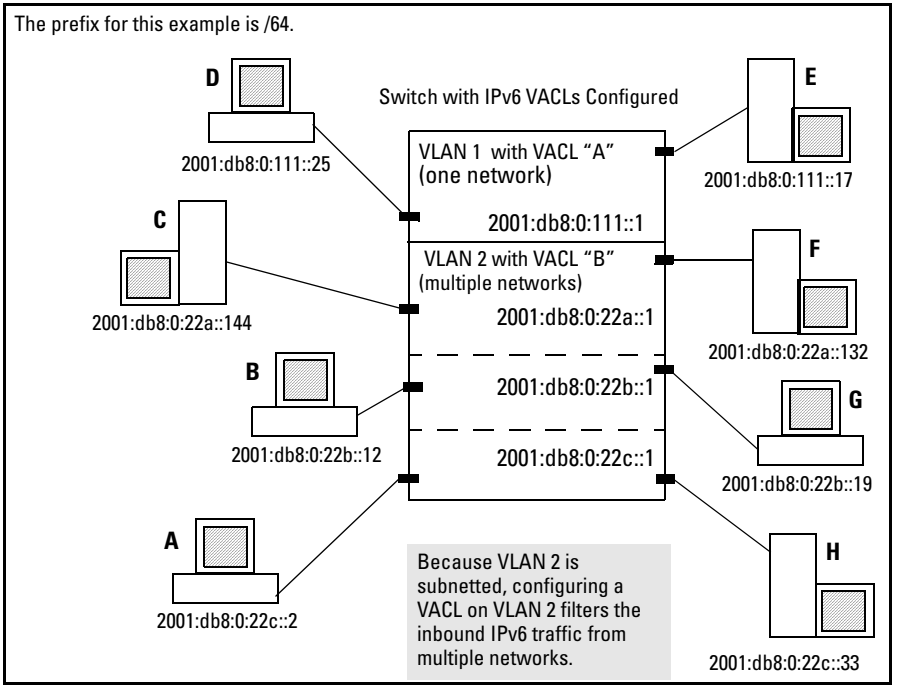

Figure 8-1. Example of VACL Filter Applications on IPv6 Traffic Entering the Switch

Note

The switch allows one IPv6 VACL assignment configured per VLAN. This is in addition to any static or RADIUS-assigned (dynamic) ACLs assigned to ports in the VLAN.

IPv6 Static Port ACL Applications

An IPv6 static port ACL filters IPv6 traffic inbound on the designated port(s).

RADIUS-Assigned (Dynamic) Port ACL Applications

Note

Beginning with software release K.14.01, IPv6 support is available for RADIUS-assigned port ACLs configured to filter inbound IPv4 and IPv6 traffic from an authenticated client. Also, the implicit deny in RADIUS-assigned ACLs applies to both IPv4 and IPv6 traffic inbound from the client. For information on enabling RADIUS-assigned ACLs, refer to the chapter titled "Configuring RADIUS Support for Switch Services" in this guide.

Dynamic (RADIUS-assigned) port ACLs are configured on RADIUS servers and can be configured to filter IPv4 and IPv6 traffic inbound from clients authenticated by such servers. For example, in figure 8-1, client "A" connects to a given port and is authenticated by a RADIUS server. Because the server is configured to assign a dynamic ACL to the port, the IPv4 and IPv6 traffic inbound on the port from client "A" is filtered. (See also "Operating Notes for IPv6 Applications" on page 8-18.)

Effect of RADIUS-Assigned ACLs When Multiple Clients Are Using the Same Port. Some network configurations may allow multiple clients to authenticate through a single port where a RADIUS server assigns a separate, RADIUS-assigned ACL in response to each client's authentication on that port. In such cases, a given client's inbound traffic will be allowed only if the RADIUS authentication response for that client includes a RADIUS-assigned ACL. Clients authenticating without receiving a RADIUS-assigned ACL will immediately be de-authenticated. For example, in figure 8-2, clients A through D authenticate through the same port (B1) on a ProCurve switch running software release K.14.01 or greater.

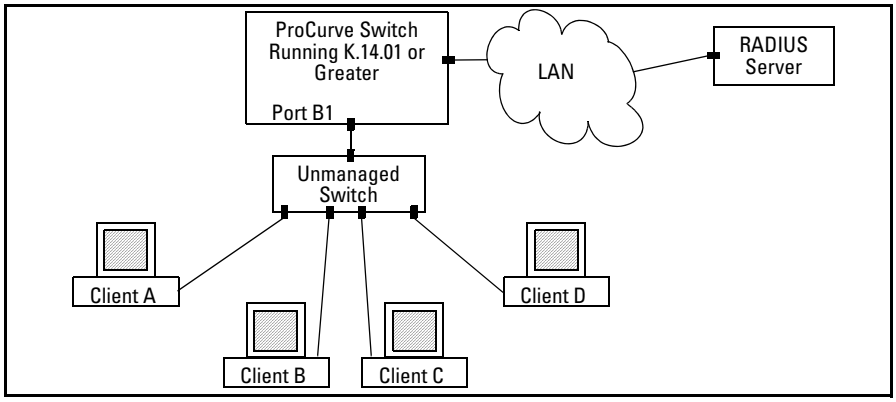

Figure 8-2. Multiple, Dual-Stack Clients Authenticating Through a Single Port

In this case, the RADIUS server must be configured to assign an ACL to port B1 for any of the authorized clients authenticating on the port.

802.1X User-Based and Port-Based Applications. *User-Based* 802.1X access control allows up to 32 individually authenticated clients on a given port. *Port-Based* access control does not set a client limit, and requires only one authenticated client to open a given port (and is recommended for applications where only one client at a time can connect to the port).

- If you configure 802.1X *user-based* security on a port and the RADIUS response includes *a RADIUS-assigned ACL* for at least one authenticated client, then the RADIUS response for *all other clients authenticated on the port* must also include a RADIUS-assigned ACL. Inbound IP traffic on the port from a client that authenticates without receiving a RADIUS-assigned ACL will be dropped and the client will be de-authenticated.

- Using 802.1X *port-based* security on a port where the RADIUS response to a client authenticating includes a RADIUS-assigned ACL, different results can occur, depending on whether any additional clients attempt to use the port and whether these other clients initiate an authentication attempt. This option is recommended for applications where only one client at a time can connect to the port, and *not recommended* for instances where multiple clients may access the same port at the same time. For more information, refer to "802.1X Port-Based Access Control" in the chapter titled "Configuring Port-Based and User-Based Access Control (802.1X)" in the latest *Access Security Guide* for your switch.

Operating Notes for IPv6 Applications.

- For RADIUS ACL applications using software release K.14.01 or greater, the switch operates in a dual-stack mode, and a RADIUS-assigned ACL filters both IPv4 and IPv6 traffic. At a minimum, a RADIUS-assigned ACL automatically includes the implicit deny for both IPv4 and IPv6 traffic. Thus, an ACL configured on a RADIUS server to filter IPv4 traffic will also deny inbound IPv6 traffic from an authenticated client unless the ACL includes ACEs that permit the desired IPv6 traffic. The reverse is true for a dynamic ACL configured on RADIUS server to filter IPv6 traffic. (ACLs are based on the MAC address of the authenticating client.) Refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the latest *Access Security Guide* for your switch.
- To support authentication of IPv6 clients:
  - The VLAN to which the port belongs must be configured with an IPv6 address.
  - Connection to an IPv6-capable RADIUS server must be supported.
- For 802.1X or MAC authentication methods, clients can authenticate regardless of their IP version (IPv4 or IPv6).
- For the Web authentication method, clients must authenticate using IPv4. However, this does not prevent the client from using a dual stack, or the port receiving a RADIUS-assigned ACL configured with ACEs to filter IPv6 traffic.
- The RADIUS server must support IPv4 and have an IPv4 address. RADIUS clients can be dual stack, IPv6-only, or IPv4-only.
- 802.1X rules for client access apply to both IPv6 and IPv4 clients for RADIUS-assigned ACLs. Refer to "802.1X User-Based and Port-Based Applications" on page 8-17.

Multiple ACL Assignments on an Interface

The switch simultaneously supports IPv6, IPv4, and RADIUS-assigned ACLs on the same interface (subject to internal resource availability). This means that traffic on a port belonging to a given VLAN "X" can simultaneously be subject to all of the ACLs listed in table 8-1.

Table 8-1. Per-Interface Multiple ACL Assignments

ACL Type	ACL Application
RADIUS-Assigned (Dynamic) ACLs	one *port-based* ACL (for first client to authenticate on the port) or up to 32 *user-based* ACLs (one per authenticated client) **Note:** If one or more user-based, RADIUS-assigned ACLs are assigned to a port, then the only traffic allowed inbound on the port is from authenticated clients.
IPv6 Static ACLs:	One static VACL for IPv6 traffic for VLAN "*X*" entering the switch through the port.
	One static port ACL for IPv6 traffic entering the switch on the port.
IPv4 Static ACLs:	one static VACL for IPv4 traffic for VLAN "X" entering the switch through the port
	one static port ACL for any IPv4 traffic entering the switch on the port
	one connection-rate ACL for inbound IPv4 traffic for VLAN "X" on the port (if the port is configured for connection-rate filtering)
	one inbound and one outbound RACL filtering routed IPv4 traffic moving through the port for VLAN "X". (Also applies to inbound, switched traffic on VLAN "X" that has a destination on the switch itself.)

Filtering Inbound Traffic with Multiple ACLS. When traffic inbound on a port is subject to multiple ACL assignments, and a RADIUS-assigned, user-based ACL is present, then this traffic must satisfy the following conditions to be permitted on the switch:

1	Originate with an authenticated client associated with the RADIUS-assigned ACL (if present).
2	Be permitted by the RADIUS-assigned ACL (if present). Includes both IPv4 and IPv6 traffic (unless the ACL is configured to exclude (drop) IPv6 traffic).
3	For IPv4-only traffic, be permitted by connection-rate ACL filtering.
4	Be permitted by a VACL configured on a VLAN to which the port is assigned.*
5	Be permitted by a PACL assigned to the port.*
6	For IPv4 traffic only, be permitted by a RACL assigned inbound to the port, if the traffic is subject to RACL rules.

*IPv4 VACLs and PACLs ignore IPv6 traffic, and the reverse.

Filtering Outbound Traffic. Outbound IPv4 traffic can be filtered only by a RACL assigned outbound on the port, and only if the traffic is subject to RACL rules. (Software version K.14.01 does not support IPv6 RACLs.)

Example of Permitting Traffic Filtered Through Multiple ACLs. On a given interface where multiple ACLs apply to the same traffic, a packet having a match with a **deny** ACE in any applicable ACL on the interface (including an implicit **deny any any**) will be dropped.

For example, suppose the following is true:

- Ports A10 and A12 belong to VLAN 100.
- A static port ACL filtering inbound IPv6 traffic is configured on port A10.
- A VACL is configured on VLAN 100.

An inbound packet entering on port A10, with a destination on port A12, will be screened by the static port ACL *and* the VACL, regardless of a match with any **permit** or **deny** action. A match with a **deny** action (including an implicit deny) in either ACL will cause the switch to drop the packet. (If the packet has a match with explicit **deny** ACEs in multiple ACLs and the log option is included in these ACEs, then a separate log event will occur for each match.)

Notes

Software release K.14.01 supports connection-rate ACLs for inbound IPv4 traffic, but not for IPv6 traffic.

Beginning with software release K.14.01, static ACL mirroring and static ACL rate-limiting are deprecated in favor of classifier-based mirroring and rate-limiting features that do not use ACLs. If ACL mirroring or ACL rate-limiting are already configured in a switch running software version K.13.*xx*, then downloading and booting from release K.14.01 or greater automatically modifies the deprecated configuration to conform to the classifier-based mirroring and rate-limiting supported in release K.14.01 or greater. For more information on this topic, refer to the chapter titled “Classifier-Based Software Configuration” in the latest *Advanced Traffic Management Guide* for your switch.

For information on traffic mirroring refer to the appendix titled “Monitoring and Analyzing Switch Operation” in the *Management and Configuration Guide* for your switch.

Features Common to All ACL Applications

- Any ACL can have multiple entries (ACEs).
- You can apply any one ACL to multiple interfaces.
- All ACEs in an ACL configured on the switch are automatically sequenced (numbered). For an existing ACL, entering an ACE without specifying a sequence number automatically places the ACE at the end of the list. Specifying a sequence number inserts the ACE into the list at the specified sequential location.
  - Automatic sequence numbering begins with "10" and increases in increments of 10. You can renumber the ACEs in an ACL and also change the sequence increment between ACEs.
  - The CLI **remark** command option allows you to enter a separate comment for each ACE.
- A source or destination IPv6 address and a prefix length, together, can define a single host, a range of hosts, or all hosts.
- Every ACL populated with one or more explicit ACEs automatically includes an Implicit Deny as the last entry in the list. The switch applies this action to packets that do not match other criteria in the ACL.
- In any ACL, you can apply an ACL log function to ACEs that have an explicit "deny" action. (The logging occurs when there is a match on a "deny" ACE that includes the **log** keyword.) The switch sends ACL logging output to Syslog, if configured, and optionally, to a console session.

You can create ACLs for the switch configuration using either the CLI or a text editor. The text-editor method is recommended when you plan to create or modify an ACL that has more entries than you can easily enter or edit using the CLI alone. Refer to "Creating or Editing ACLs Offline" on page 8-87.

General Steps for Planning and Configuring ACLs

1. Identify the ACL action to apply. As part of this step, determine the best points at which to apply specific ACL controls. For example, you can improve network performance by filtering unwanted IPv6 traffic at the edge of the network instead of in the core. Also, on the switch itself, you can improve performance by filtering unwanted IPv6 traffic where it is inbound to the switch instead of outbound.

Traffic Source	ACL Application
IPv6 traffic from a specific, authenticated client	RADIUS-assigned ACL for inbound IPv6 traffic from an authenticated client on a port*
IPv6 traffic entering the switch on a specific port	static port ACL (static-port assigned) for inbound IPv6 traffic on a port from any source
IPv6 traffic entering the switch on a specific VLAN	VACL (VLAN ACL)

*For more on this option, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the latest version of the *Access Security Guide* for your switch. Refer also to the documentation for your RADIUS server.

2. Identify the IPv6 traffic types to filter:
   - The SA and/or the DA of IPv6 traffic you want to permit or deny. This can be a single host, a group of hosts, a subnet, or all hosts.
   - IPv6 traffic of a specific protocol type (0-255)
   - TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
   - UDP traffic (only) or UDP traffic for a specific UDP port
   - ICMP traffic (only) or ICMP traffic of a specific type and code
   - Any of the above with specific DSCP settings
3. Design the ACLs for the control points (interfaces) you have selected. Where you are using explicit "deny" ACEs, you can optionally use the ACL logging feature for notification that the switch is denying unwanted packets.
4. Configure the ACLs on the selected switches.
5. Assign the ACLs to the interfaces you want to filter, using the ACL application (static port ACL or VACL) appropriate for each assignment. (For RADIUS-assigned ACLs, refer to the footnote in the table in step 1 on page 8-22.)

6. Test for desired results.

For more details on ACL planning considerations, refer to “Planning an ACL Application” on page 8-28.

IPv6 ACL Operation

Introduction

An ACL is a list of one or more Access Control Entries (ACEs), where each ACE consists of a matching criteria and an action (permit or deny). An ACL applies only to the switch in which it is configured. ACLs operate on assigned interfaces, and offer these traffic filtering options:

- IPv6 traffic inbound on a port.
- IPv6 traffic inbound on a VLAN.

The following table lists the range of interface options:

Interface	ACL Application	Application Point	Filter Action
Port	Static Port ACL (switch configured)	inbound on the switch port	inbound IPv6 traffic
	RADIUS-assigned ACL[1]	inbound on the switch port used by authenticated client	inbound IPv6 traffic from the authenticated client
VLAN	VACL	entering the switch on the VLAN	inbound IPv6 traffic

[1]This chapter describes ACLs statically configured on the switch. For information on RADIUS-assigned ACLs, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services"in the latest version of the *Access Security Guide* for your switch.

Note

After you assign an ACL to an interface, the default action on the interface is to implicitly deny any IPv6 traffic that is not specifically permitted by the ACL. (This applies only in the direction of traffic flow filtered by the ACL.)

The Packet-filtering Process

Sequential Comparison and Action. When an ACL filters a packet, it sequentially compares each ACE's filtering criteria to the corresponding data in the packet until it finds a match. The action indicated by the matching ACE (deny or permit) is then performed on the packet.

Implicit Deny. If a packet does not have a match with the criteria in any of the ACEs in the ACL, the ACL denies (drops) the packet. If you need to override the implicit deny so that a packet that does not have a match will be permitted, then configure **permit ipv6 any any** as the last ACE in the ACL. This directs the ACL to permit (forward) packets that do not have a match with any earlier ACE listed in the ACL, and prevents these packets from being filtered by the implicit **deny ipv6 any any**.

Example. Suppose the ACL in figure 8-3 is assigned to filter the IPv6 traffic from an authenticated client on a given port in the switch:

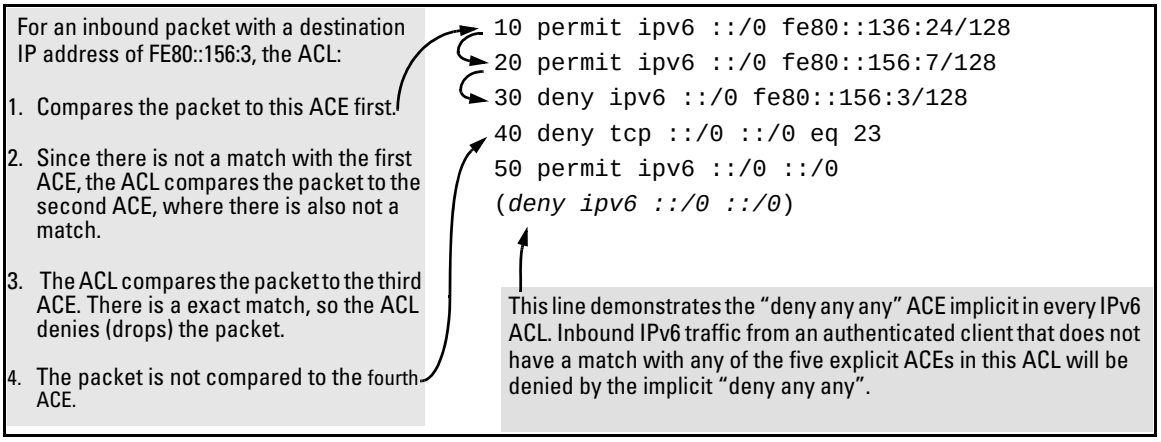
For an inbound packet with a destination IP address of FE80::156:3, the ACL:
1. Compares the packet to this ACE first.
2. Since there is not a match with the first ACE, the ACL compares the packet to the second ACE, where there is also not a match.
3. The ACL compares the packet to the third ACE. There is a exact match, so the ACL denies (drops) the packet.
4. The packet is not compared to the fourth ACE.
10 permit ipv6 ::/0 fe80::136:24/128
20 permit ipv6 ::/0 fe80::156:7/128
30 deny ipv6 ::/0 fe80::156:3/128
40 deny tcp ::/0 ::/0 eq 23
50 permit ipv6 ::/0 ::/0
(deny ipv6 ::/0 ::/0)
This line demonstrates the "deny any any" ACE implicit in every IPv6 ACL. Inbound IPv6 traffic from an authenticated client that does not have a match with any of the five explicit ACEs in this ACL will be denied by the implicit "deny any any".

Figure 8-3. Example of Sequential Comparison

As shown above, the ACL tries to apply the first ACE in the list. If there is not a match, it tries the second ACE, and so on. When a match is found, the ACL invokes the configured action for that entry (permit or drop the packet) and no further comparisons of the packet are made with the remaining ACEs in the list. This means that when an ACE whose criteria matches a packet is found, the action configured for that ACE is invoked, and any remaining ACEs in the ACL are ignored. *Because of this sequential processing, successfully*

implementing an ACL depends in part on configuring ACEs in the correct order for the overall policy you want the ACL to enforce.

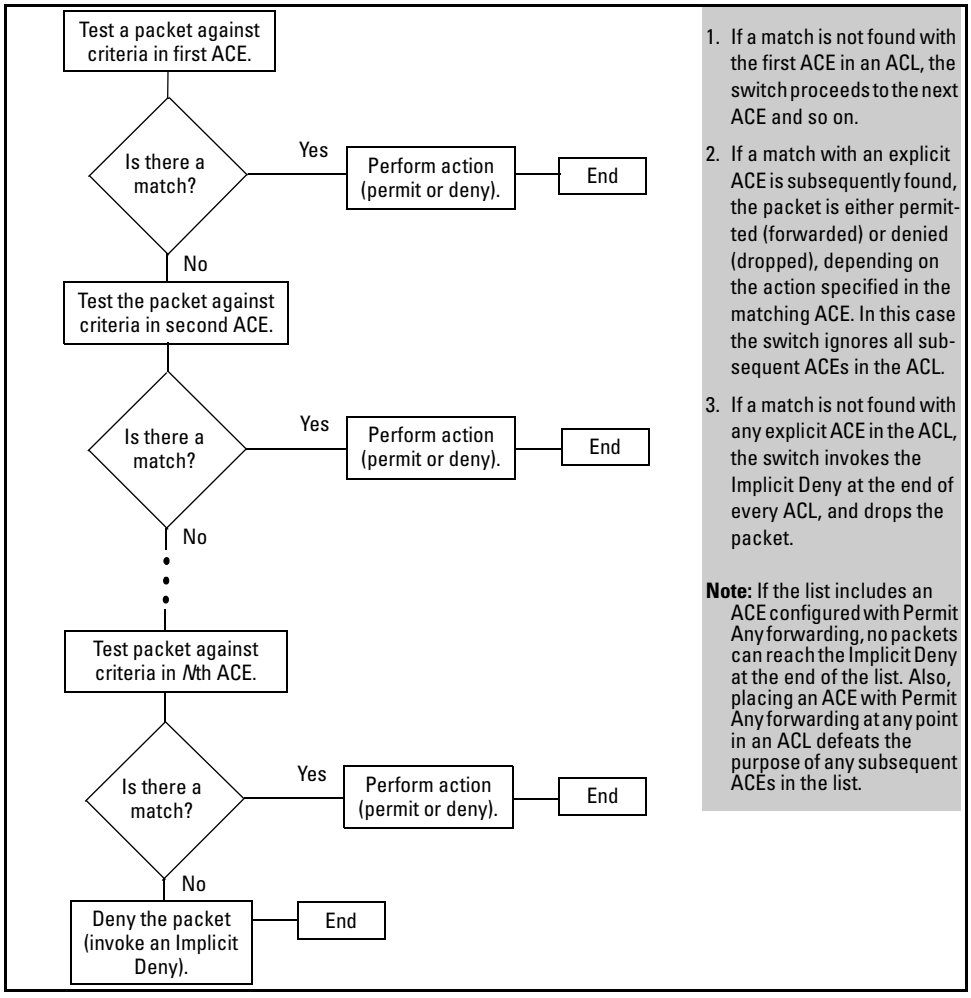
Test a packet against criteria in first ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test the packet against criteria in second ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Test packet against criteria in Nth ACE.
Is there a match?
Yes
Perform action (permit or deny).
End
No
Deny the packet (invoke an Implicit Deny).
End
1. If a match is not found with the first ACE in an ACL, the switch proceeds to the next ACE and so on.
2. If a match with an explicit ACE is subsequently found, the packet is either permitted (forwarded) or denied (dropped), depending on the action specified in the matching ACE. In this case the switch ignores all subsequent ACEs in the ACL.
3. If a match is not found with any explicit ACE in the ACL, the switch invokes the Implicit Deny at the end of every ACL, and drops the packet.
Note: If the list includes an ACE configured with Permit Any forwarding, no packets can reach the Implicit Deny at the end of the list. Also, placing an ACE with Permit Any forwarding at any point in an ACL defeats the purpose of any subsequent ACEs in the list.

Figure 8-4. The Packet-Filtering Process in an ACL with *N* Entries (ACEs)

Note

The order in which an ACE occurs in an ACL is significant. For example, if an ACL contains six ACEs, but the first ACE allows "Permit Any" forwarding, then the ACL permits all IPv6 traffic, and the remaining ACEs in the list do not apply, even if they have a match with any traffic permitted by the first ACE.

For example, suppose you want to configure an ACL (with an ID of "Test-02") to invoke these policies for IPv6 traffic entering the switch on VLAN 100:

1. Permit inbound IPv6 traffic from 2001:db8:0:fb::11:42.
2. Deny only the inbound Telnet traffic from 2001:db8:0:fb::11:101.
3. Permit inbound IPv6 traffic from 2001:db8:0:fb::11:101.
4. Permit only inbound Telnet traffic from 2001:db8:0:fb::11:33.
5. Deny any other inbound IPv6 traffic.

The following ACL, when assigned to filter inbound traffic on VLAN 100, supports the above case:

```
ipv6 access-list "Test-02"

1 10 permit ipv6 2001:db8:0:fb::11:42/128 ::/0

2 20 deny tcp 2001:db8:0:fb::11:101/128 eq 23 ::/0

3 30 permit ipv6 2001:db8:0:fb::11:101/128 ::/0

4 40 permit tcp 2001:db8:0:fb::11:33/128 ::/0 eq 23

5 < Implicit Deny Any Any >
```

1. **Permits** IPv6 traffic from 2001:db8:0:fb::11:42. Packets matching this criterion are permitted and will not be compared to any later ACE in the list. Packets not matching this criterion will be compared to the next entry in the list.
2. **Denies** IPv6 Telnet traffic from 2001:db8:0:fb::11:101. Packets matching this criterion are dropped and are not compared to later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
3. **Permits** IPv6 traffic from 2001:db8:0:fb::11:101. Packets matching this criterion will be permitted and will not be compared to any later criteria in the list. Because this entry comes after the entry blocking Telnet traffic from this same address, there will not be any Telnet packets to compare with this entry; they have already been dropped as a result of matching the preceding entry.
4. **Permits** IPv6 Telnet traffic from 2001:db8:0:fb::11:33. Packets matching this criterion are permitted and are not compared to any later criteria in the list. Packets not matching this criterion are compared to the next entry in the list.
5. This entry does not appear in an actual ACL, but is implicit as the last entry in every IPv6 ACL. Any IPv6 packets that do not match any of the criteria in the preceding ACL entries will be denied (dropped) from the VLAN.

Figure 8-5. Example of How an ACL Filters Packets

To assign the above ACL, you would use this command:

```
ProCurve(config)# vlan 100 ipv6 access-group Test-02 vlan
```

It is important to remember that ACLs configurable on the switch include an implicit **deny ipv6 any any**. That is, IPv6 packets that the ACL does not *explicitly* permit or deny will be *implicitly* denied, and therefore dropped instead of forwarded on the interface. If you want to preempt the implicit deny so that packets not explicitly denied by other ACEs in the ACL will be permitted,

insert an explicit **permit ipv6 any any** as the last ACE in the ACL. Doing so permits any packet not explicitly denied by earlier entries. (Note that this solution would not apply in the preceding example, where the intention is for the switch to forward only the explicitly permitted packets entering the switch on VLAN 100.)

Planning an ACL Application

Before creating and implementing ACLs, define the policies you want your ACLs to enforce, and understand how the ACL assignments will impact your network users.

Note

IPv6 traffic entering the switch on a given interface is filtered by the ACLs configured for inbound traffic on that interface. For this reason, an inbound packet will be denied (dropped) if it has a match with an implicit (or explicit) **deny ipv6 any any** in *any* of the inbound ACLs applied to the interface.

(Refer to “Multiple ACL Assignments on an Interface” on page 8-18.)

IPv6 Traffic Management and Improved Network Performance

You can use ACLs to block IPv6 traffic from individual hosts, workgroups, or subnets, and to block access to VLANs, subnets, devices, and services. Traffic criteria for ACLs include:

- Switched IPv6 traffic
- IPv6 traffic of a specific protocol type (0-255)
- TCP traffic (only) for a specific TCP port or range of ports, including optional control of connection traffic based on whether the initial request should be allowed
- UDP traffic (only) or UDP traffic for a specific UDP port
- ICMP traffic (only) or ICMP traffic of a specific type and code
- Any of the above with specific precedence and/or ToS settings

Depending on the source and/or destination of a given IPv6 traffic type, you must also determine the ACL application(s) (VACL or static port ACL) needed to filter the traffic on the applicable switch interfaces. Answering the following questions can help you to design and properly position ACLs for optimum network usage.

- What are the logical points for minimizing unwanted IPv6 traffic, and what ACL application(s) should be used? In many cases it makes sense to prevent unwanted IPv6 traffic from reaching the core of your network by configuring ACLs to drop unwanted IPv6 traffic at or close to the edge of the network. (The earlier in the network path you can deny unwanted traffic, the greater the benefit for network performance.)
- From where is the traffic coming? The source and destination of IPv6 traffic you want to filter determines the ACL application to use (VACL, static port ACL, and RADIUS-assigned ACL).
- What IPv6 traffic should you explicitly deny? Depending on your network size and the access requirements of individual hosts, this can involve creating a large number of ACEs in a given ACL (or a large number of ACLs), which increases the complexity of your solution.
- What IPv6 traffic can you implicitly deny by taking advantage of the implicit **deny ipv6 any any** to deny IPv6 traffic that you have not explicitly permitted? This can reduce the number of entries needed in an ACL.
- What IPv6 traffic should you permit? In some cases you will need to explicitly identify permitted IPv6 traffic. In other cases, depending on your policies, you can insert an ACE with “permit any” forwarding at the end of an ACL. This means that IPv6 traffic not specifically matched by earlier entries in the list will be permitted.

Security

ACLs can enhance security by blocking IPv6 traffic carrying an unauthorized source IPv6 address (SA). This can include:

- blocking access from specific devices or interfaces (port or VLAN)
- blocking access to or from subnets in your network
- blocking access to or from the internet
- blocking access to sensitive data storage or restricted equipment

- preventing specific TCP, UDP, and ICMP traffic types, including unauthorized access using functions such as Telnet, SSH, and web browser

You can also enhance switch management security by using ACLs to block IPv6 traffic that has the switch itself as the destination address (DA).

Caution

ACLs can enhance network security by denying selected IPv6 traffic, and can serve as one aspect of maintaining network security. *However, because ACLs do not provide user or device authentication, or protection from malicious manipulation of data carried in IPv6 packet transmissions, they should not be relied upon for a complete security solution.*

Note

ACLs in the switches covered by this guide do not filter non-IPv6 traffic such as IPv4, AppleTalk, and IPX packets.

Guidelines for Planning the Structure of an ACL

After determining the ACL application (VACL or static port ACL) to use at a particular point in your network, determine the order in which to apply individual ACEs to filter IPv6 traffic. (For information on ACL applications, refer to "IPv6 ACL Applications" on page 8-13.).

- The sequence of ACEs is significant. When the switch uses an ACL to determine whether to permit or deny a packet on a particular VLAN, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet.
- The first match in an ACL dictates the action on a packet. Subsequent matches in the same ACL are ignored. However, if a packet is permitted by one ACL assigned to an interface, but denied by another ACL assigned to the same interface, the packet will be denied on the interface.
- On any ACL, the switch implicitly denies IPv6 packets that are not explicitly permitted or denied by the ACEs configured in the ACL. If you want the switch to forward a packet for which there is not a match

in an ACL, append an ACE that enables Permit Any forwarding as the last ACE in an ACL. This ensures that no packets reach the Implicit Deny case for that ACL.

- Generally, you should list ACEs from the most specific (individual hosts) to the most general (subnets or groups of subnets) unless doing so permits IPv6 traffic that you want dropped. For example, an ACE allowing a series of workstations to use a specialized printer should occur earlier in an ACL than an entry used to block widespread access to the same printer.

ACL Configuration and Operating Rules

- **VACLs:** A VACL filters IPv6 traffic entering the switch on the VLAN(s) to which it is assigned.
- **Static Port ACLs:** A static port ACL filters IPv6 traffic entering the switch on the port(s) or trunk(s) to which it is assigned.
- **Per Switch ACL Limits for All ACL Types.** At a minimum an ACL must have one, explicit "permit" or "deny" Access Control Entry. You can configure up to 2048 ACLs (IPv4 and IPv6 combined). Total ACEs in all ACLs depends on the combined resource usage by ACL and other features (For more on this topic, refer to "Monitoring Shared Resources" on page 8-105.)
- **Implicit Deny:** In any static ACL, the switch implicitly (automatically) applies an implicit **deny ipv6 any any** that does not appear in **show** listings. This means that the ACL denies any packet it encounters that does not have a match with an entry in the ACL. Thus, if you want an ACL to permit any IPv6 packets that you have not expressly denied, you must enter a **permit ipv6 any any** as the last ACE in an ACL. Because, for a given packet, the switch sequentially applies the ACEs in an ACL until it finds a match, any packet that reaches a **permit ipv6 any any** entry will be permitted, and will not encounter the implicit "Deny" ACE the switch automatically includes at the end of the ACL. For an example, refer to figure 8-9 on page 8-40. For implicit deny operation in RADIUS-assigned (dynamic) ACLs, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the latest *Access Security Guide* for your Switch.
- **Explicitly Permitting IPv6 Traffic:** Entering a **permit ipv6 any any** ACE in an ACL permits the IPv6 traffic not previously permitted or denied by that ACL. Any ACEs listed after that point do not have any effect.

- **Explicitly Denying IPv6 Traffic:** Entering a **deny ipv6 any any** ACE in an ACL denies IPv6 traffic not previously permitted or denied by that ACL. Any ACEs listed after that point have no effect.
- **Replacing One ACL with Another of the Same Type:** For a specific interface, the most recent ACL assignment using a given application replaces any previous ACL assignment using the same application on the same interface. For example, if you assigned a VACL named "Test-01" to filter inbound IPv6 traffic on VLAN 20, but later, you assigned another VACL named "Test-02" to filter inbound IPv6 traffic on this same VLAN, VACL "Test-02" replaces VACL "Test-01" as the ACL to use.
- **Static Port ACLs:** These are applied per-port, per port-list, or per static trunk. Adding a port to a trunk applies the trunk's ACL configuration to the new member. If a port is configured with an ACL, the ACL must be removed before the port is added to the trunk. Also, removing a port from an ACL-configured trunk removes the ACL configuration from that port.
- **VACLs:** These filter IPv6 traffic entering the switch through any port belonging to the designated VLAN. VACLs do not filter IPv6 traffic leaving the switch.
- **VACLs Operate On Static VLANs:** You can assign an ACL to any VLAN that is statically configured on the switch. ACLs do not operate with dynamic VLANs.
- **A VACL Affects All Physical Ports in a Static VLAN:** A VACL assigned to a VLAN applies to all physical ports on the switch belonging to that VLAN, including ports that have dynamically joined the VLAN.

How an ACE Uses a Prefix To Screen Packets for SA and DA Matches

For an IPv6 ACL, a match with a packet occurs when both the protocol and the SA/DA configured in a given ACE within the ACL are a match with the same criteria in a packet being filtered by the ACL.

In IPv6 ACEs, prefixes define how many leading bits in the SA and DA to use for determining a match. That is, the switch uses IPv6 prefixes in CIDR format to specify how many leading bits in a packet's SA and DA must be an exact match with the same bits in an ACE. The bits to the right of the prefix are "wildcards", and are not used to determine a match.

Prefix	Range of Applicable Addresses	Examples
/0	any IPv6 host	::/0
/ 1 — /127	all IPv6 hosts within the range defined by the number of bits in the prefix	2001:db8::/48 2001:db8::/64
/128	one IPv6 host	2001:db8::218:71ff:fec4:2f00/128

For example, the following ACE applies to Telnet packets from a source address where the leading bits are set to 2001:db8:10:1 and any destination address where the leading bits are set to 2001:db8:10:1:218:71ff:fec.

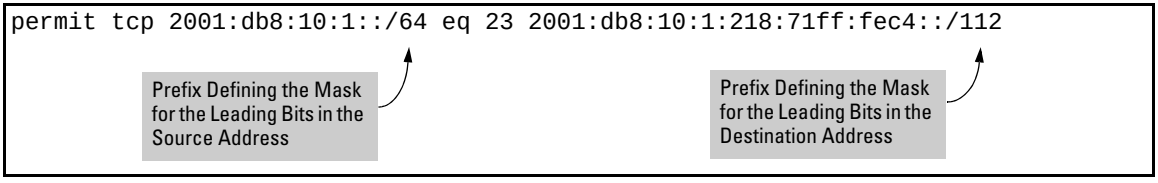
permit tcp 2001:db8:10:1::/64 eq 23 2001:db8:10:1:218:71ff:fec4::/112
Prefix Defining the Mask for the Leading Bits in the Source Address
Prefix Defining the Mask for the Leading Bits in the Destination Address

Figure 8-6. Example of SA/DA Prefix Lengths

Thus, in the above example, if an IPv6 telnet packet has an SA match with the ACE's leftmost 64 bits and a DA match with the ACE's leftmost 112 bits, then there is a match and the packet is permitted. In this case, the source and destination addresses allowed are:

Address	Prefix	Range of Unicast Addresses
Source (SA)	2001:db8:10:1	< *prefix* >::0 to < *prefix* >:FFFF:FFFF:FFFF:FFFF
Destination (DA)	2001:db8:10:1:218:71ff:fec4	< *prefix* >:0 to < *prefix* >:FFFF

To summarize, when the switch compares an IPv6 packet to an ACE in an ACL, it uses the subnet prefixes configured with the SA and DA in the ACE to determine how many leftmost, contiguous bits in the ACE's SA and DA must be matched by the same bits in the SA and DA carried by the packet. Thus, the subnet prefixes specified with the SA and DA in an ACE determine the ranges of source and destination addresses acceptable for a match between the ACE and a packet being filtered.

Prefix Usage Differences Between ACLs and Other IPv6 Addressing

For ACLs, the prefix is used to specify the leftmost bits in an address that are meaningful for a packet match. In other ACL usage, the prefix separates network and subnet values from the device identifier in an address.

Prefix Usage	Examples	Notes
For an SA or DA in the ACE belonging to an IPv6 ACL, the associated prefix specifies how many consecutive, leading bits in the address are used to define a match with the corresponding bits in the SA or DA of a packet being filtered.	2620:0:a03:e102:215:60ff:fe7a:adc0/128	All bits. Used for a specific SA or DA.
	2620:0:a03:e102:215/80	The first 80 bits. Used for an SA or DA having 2620:0:a03:e102:215 in the leftmost 80 bits of an address.
	::/0	Zero bits. Used to allow a match with "Any" SA or DA.
For the IPv6 address assigned to a given device, the prefix defines the type of address and the network and subnet in which the address resides. In this case, the bits to the right of the prefix comprise the device identifier.	fe80::215:60ff:fe7a:adc0/64	Link-Local address with a prefix of 64 bits and a device ID of 64 bits.
	2620:0:a03:e102:215:60ff:fe7a:adc0/64	Global unicast address with a prefix of 64 bits and a device ID of 64 bits.

Configuring and Assigning an IPv6 ACL

ACL Feature	Page
Adding or Removing an ACL	8-62
Enabling or Disabling ACL Filtering	8-65

General Steps for Implementing IPv6 ACLs

1. Configure one or more ACLs. This creates and stores the ACL(s) in the switch configuration.
2. Assign an ACL. This step uses one of the following applications to assign the ACL to an interface:
   - VACL (IPv6 traffic entering the switch on a given VLAN)
   - Static Port ACL (IPv6 traffic entering the switch on a given port, port list, or static trunk)

Permit/Deny Options

You can use the following criteria as options for permitting or denying a packet:

- source IPv6 address
- destination IPv6 address
- IPv6 protocol options:
  - all IPv6 traffic
  - IPv6 traffic of a specific protocol type (0-255)
  - IPv6 traffic for a specific TCP port or range of ports, including:
    - optional control of connection (established) traffic based on whether the initial request should be allowed
    - TCP flag (control bit) options
  - IPv6 traffic for a specific UDP port or range of ports
  - IPv6 traffic for a specific ICMP type and code
  - any of the above with specific DSCP precedence or ToS settings

Carefully plan ACL applications before configuring specific ACLs. For more on this topic, refer to “Planning an ACL Application” on page 8-28.

ACL Configuration

After you enter an ACL command, you may want to inspect the resulting configuration. This is especially true where you are entering multiple ACEs into an ACL. Also, it is helpful to understand the configuration structure when using later sections in this chapter.

The basic ACL structure includes four elements:

1. ACL identity: This is a string of up to 64 characters specifying the ACL name.
2. Optional **remark** entries.

3. One or more deny/permit list entries (ACEs): One entry per line.

Element	Notes
Identifier	Alphanumeric; Up to 64 Characters, Including Spaces
Remark	Allows up to 100 alphanumeric characters, including blank spaces. (If any spaces are used, the remark must be enclosed in a pair of single or double quotes.) A remark is associated with a particular ACE and will have the same sequence number as the ACE. (One remark is allowed per ACE.) Refer to "Attaching a Remark to an ACE" on page 8-73.
Maximum ACEs Per Switch	The maximumnumber of ACEs supported by the switch is up to 3072 for IPv6 ACEs and up to 3072 for IPv4 ACEs. The maximum number of ACEs applied to a VLAN or port depends on the concurrent resource usage by multiple configured features. For more information, use the **show < qos | access-list > resources** command and/or refer to "Monitoring Shared Resources" on page 8-105.

4. Implicit Deny: Where an ACL is applied to an interface, it denies any packets that do not have a match with any of the ACEs explicitly configured in the list. The Implicit Deny does not appear in ACL configuration listings, but always functions when the switch uses an ACL to filter packets. (You cannot delete the Implicit Deny, but you can supersede it with a **permit ipv6 any any** ACE.)

ACL Configuration Structure

Individual ACEs in an IPv6 ACL include:

- Optional remark statements
- A permit/deny statement
- Source and destination IPv6 addressing
- Choice of IPv6 criteria
- Optional ACL **log** command (for **deny** entries)

```
ipv6 access-list < identifier >
    [ seq-# ]
        [ remark < remark-str ]
        < permit | deny >
            0 - 255
            esp
            ah
            sctp
            icmp
                < SA > [operator < value >]
                < DA > [operator < value >] [type [code] | icmp-msg ]     [dscp < codepoint | precedence >]
            ipv6
            tcp
                < SA > [operator < value >]
                < DA > [operator < value >]
                    [dscp < codepoint | precedence]
                    [established]
                    [ack | fin | rst | syn]
            udp
                < SA > [operator < value >]
                < DA > [operator < value >] [dscp < codepoint | precedence ]
        [log]  (Allowed only with "deny" ACEs.)
    . . .
    < Implicit Deny Any Any >
    exit
```

Figure 8-7. General Structure Options for an IPv6 ACL

For example, the ACL in figure 8-8 filters traffic for individual hosts in some instances and all hosts in others:

```
ProCurve# show run
  .
  .
  .
ipv6 access-list "Sample-List-1"
   10 permit ipv6 2001:db8:0:130::55/128 2001:db8:0:130::240/128
   20 permit tcp ::/0 ::/0 eq 23
   30 remark "ALLOWS HTTP FROM SINGLE HOST."
   30 permit tcp 2001:db8:0:140::14/128 eq 80 ::/0 eq 3871
   40 remark "DENIES HTTP FROM ANY TO ANY."
   40 deny tcp ::/0 ::/0 eq 80 log
   50 deny udp 2001:db8:0:150::44/128 eq 69 2001:db8:0:120::19/128
              range 3680 3690 log
   60 deny udp ::/0 2001:db8:0:150::121/128 log
   70 permit ipv6 2001:db8:0:01::/56 ::/0
   exit
```

Figure 8-8. Example of a Displayed ACL Configuration

Line	Action
10	Permits all IPv6 traffic from the host at 2001:db8:0:130::55 to the host at 2001:db8:0:130::240.
20	Permits all Telnet traffic from any source to any destination.
30	Includes a remark and permits TCP port80 traffic received at any destination as port 3871 traffic.
40	Includes a remark and denies TCP port 80traffic received at any destination, and causes a log message to be generated when a match occurs.
50	Denies UDP port 69 (TFTP) traffic sentfrom the host at 2001:db8:0:150::44 to the host at 2001:db8:0:120::19 with a destination port number in the range of 3680 - 3690, and causes a log message to be generated when a match occurs.
60	Denies UDP traffic from any source to the host at 2001db8:0:150::121, and causes a log message to be generated when a match occurs.
70	Permits all IPv6 traffic with an SA prefixof 2001:db8:0:01/56 that is not already permitted or denied by the preceding ACEs in the ACL.
Note: An implicit "Deny IPv6 any any" is automatically applied following the last line (70, in this case), and denies all IPv6 traffic not already permitted or denied by the ACEs in lines 10 through 70.	

ACL Configuration Factors

The Sequence of Entries in an ACL Is Significant

When the switch uses an ACL to determine whether to permit or deny a packet, it compares the packet to the criteria specified in the individual Access Control Entries (ACEs) in the ACL, beginning with the first ACE in the list and proceeding sequentially until a match is found. When a match is found, the switch applies the indicated action (permit or deny) to the packet. This is significant because, once a match is found for a packet, subsequent ACEs in the same ACL will not be applied to that packet, regardless of whether they match the packet.

For example, suppose that you have applied the ACL shown in figure 8-9 to inbound IPv6 traffic on VLAN 1 (the default VLAN):

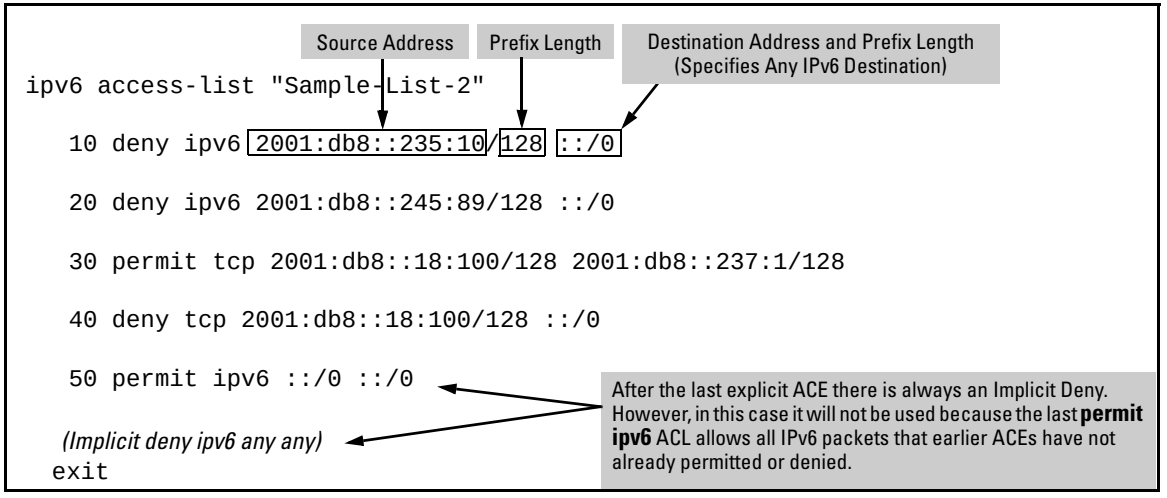
Source Address
Prefix Length
Destination Address and Prefix Length
(Specifies Any IPv6 Destination)
ipv6 access-list "Sample-List-2"
10 deny ipv6 2001:db8::235:10/128 ::/0
20 deny ipv6 2001:db8::245:89/128 ::/0
30 permit tcp 2001:db8::18:100/128 2001:db8::237:1/128
40 deny tcp 2001:db8::18:100/128 ::/0
50 permit ipv6 ::/0 ::/0
(Implicit deny ipv6 any any)
exit
After the last explicit ACE there is always an Implicit Deny. However, in this case it will not be used because the last permit ipv6 ACL allows all IPv6 packets that earlier ACEs have not already permitted or denied.

Figure 8-9. Example of an ACE that Permits All IPv6 Traffic Not Implicitly Denied

Table 8-2. Effect of the Above ACL on Inbound IPv6 Traffic in the Assigned VLAN

Line #	Action
n/a	Shows IP type (IPv6) and ID (Sample-List-2).
10	A packet from source address 2001:db8:235:10 will be denied (dropped). This ACE filters out all packets received from 2001:db8:235:10. As a result, IPv6 traffic from that device will not be allowed and packets from that device will not be compared against any later entries in the list.
20	A packet from IPv6 source address 2001:db8::245:89 will be denied (dropped). This ACE filters out all packets received from 2001:db8::245:89. As the result, IPv6 traffic from that device will not be allowed and packets from that device will not be compared against any later entries in the list.
30	A TCP packet from SA 2001:db8::18:100 with a DA of 2001:db8::237:1 will be permitted (forwarded). Since no earlier ACEs in the list have filtered TCP packets from 2001:db8::18:100 with a destination of 2001:db8::237:1, the switch will use this ACE to evaluate such packets. Any packets that meet this criteria will be forwarded. (Any packets that do not meet this TCP source-destination criteria are not affected by this ACE.)
40	A TCP packet from source address 2001:db8::18:100 to ***any*** destination address will be denied (dropped). Since, in this example, the intent is to block TCP traffic from 2001:db8::18:100 to any destination ***except*** the destination stated in the ACE at line 30, this ACE must follow the ACE at line 30. (If their relative positions were exchanged, all TCP traffic from 2001:db8::18:100 would be dropped, including the traffic for the 2001:db8::237:1 destination.)
50	Any packet from any IPv6 source address to any IPv6 destination address will be permitted (forwarded). The only traffic filtered by this ACE will be packets not specifically permitted or denied by the earlier ACEs.
n/a	The *Implicit Deny (***deny ipv6 any any***)* is a function the switch automatically adds as the last action in all IPv6 ACLs. It denies (drops) traffic from any source to any destination that has not found a match with earlier entries in the ACL. In this example, the ACE at line 50 permits (forwards) any traffic not already permitted or denied by the earlier entries in the list, so there is no traffic remaining for action by the Implicit Deny function.
exit	Defines the end of the ACL.

Allowing for the Implied Deny Function

In any ACL having one or more ACEs there will always be a packet match. This is because the switch automatically applies the Implicit Deny as the last ACE in any ACL. This function is not visible in ACL listings, but is always present. (Refer to figure 8-9.) This means that if you configure the switch to use an ACL for filtering either inbound or outbound traffic on a VLAN, any IPv6 packets not specifically permitted or denied by the explicit entries you create will be denied by the Implicit Deny action. If you want to preempt the Implicit Deny (so that IPv6 traffic not specifically addressed by earlier ACEs in a given ACL will be permitted), insert an explicit **permit ipv6 any any** as the last explicit ACE in the ACL.

A Configured ACL Has No Effect Until You Apply It to an Interface

The switch stores ACLs in the configuration file. Until you actually assign an ACL to an interface, it is present in the configuration, but not used (and does not use any of the monitored resources described in the appendix titled "Monitored Resources" in the latest version of the *Management and Configuration Guide* for your switch.)

You Can Assign an ACL Name to an Interface Even if the ACL Has Not Been Configured

In this case, if you subsequently create an ACL with that name, the switch automatically applies each ACE as soon as you enter it in the running-config file. Similarly, if you modify an existing ACE in an ACL you already applied to an interface, the switch automatically implements the new ACE as soon as you enter it. (See "General ACL Operating Notes" on page 8-105.) The switch allows up to 2048 ACLs each for IPv4 and IPv6 For example, if you configure two ACLs, but assign only one of them to a VLAN, the ACL total is two, for the two unique ACL names. If you then assign the name of an empty ACL to a VLAN, the new ACL total is three, because the switch now has three unique ACL names in its configuration. (RADIUS-based ACL resources are drawn from the IPv4 allocation).

(For information on switch resource use, refer to "Monitoring Shared Resources" on page 8-105.)

Using the CLI To Create an ACL

Command	Page
access-list	8-43

You can use either the switch CLI or an offline text editor to create an ACL. This section describes the CLI method, which is recommended for creating short ACLs. (To use the offline method, refer to "Creating or Editing ACLs Offline" on page 8-87.)

General ACE Rules

These rules apply to all ACEs you create or edit using the CLI:

Adding or Inserting an ACE in an ACL. To *add* an ACE to the end of an ACL, use the **ipv6 access-list < *name-str* >** command to enter the context for a specific IPv6 ACL. (If the ACL does not already exist in the switch configuration, this command creates it.) Then enter the text of the ACE without specifying a sequence number. For example, the following pair of commands enter the context of an ACL named "List-1" and add a "permit" ACE to the end of the list. This new ACE permits the IPv6 traffic from the device at 2001:db8:0:a9:8d:100 to go to all destinations.

```
ProCurve(config)# ipv6 access-list List-1
ProCurve(config-ipv6-acl)# permit host 2001:db8:0:a9::8d:100 any
```

To insert an ACE anywhere in an existing ACL, enter the context of the ACL and specify a sequence number. For example, to insert a new ACE as line 15 between lines 10 and 20 in an existing ACL named "List-2" to deny traffic from the device at 2001:db8:0:a9::8d:77, you would use the following commands:

```
ProCurve(config)# ipv6 access-list List-2
ProCurve(config-ipv6-acl)# 15 deny ipv6 host 2001:db8:0:a9::8d:77 any
```

To Delete an ACE. Enter the ACL context and delete the sequence number for the unwanted ACE. (To view the sequence numbers of the ACEs in a list, use **show access-list < *acl-name-str* > config**.) For example, to delete the ACE at line 40 in an ACL named "List-2", you would enter the following commands:

```
ProCurve(config)# ipv6 access-list List-2 config
ProCurve(config-ipv6-acl)# no 40
```

Duplicate ACE Sequence Numbers. Duplicate sequence numbering for ACEs are not allowed in the same ACL. Attempting to enter a duplicate ACE displays the **Duplicate sequence number** message.

Using CIDR Notation To Enter the IPv6 ACL Prefix Length

CIDR (Classless Inter-Domain Routing) notation is used to specify ACL prefix lengths. The switch compares the address bits specified by a prefix length for an SA or DA in an ACE with the corresponding address bits in a packet being filtered by the ACE. If the designated bits in the ACE and in the packet have identical settings, then the addresses match.

Table 8-3. Examples of CIDR Notation for Prefix Lengths

SA or DA Used In an ACL with CIDR Notation	Resulting Prefix Length Defining an Address Match	Meaning
2620:0:a03:e102::/64	2620:0:a03:e102	The leftmost 64 bits must match. The remaining 64 bits are wildcards.
2620:0:a03:e102:215::/80	2620:0:a03:e102:215	The leftmost 80 bits must match. The remaining 48 bits are wildcards.
2620:0:a03:e102:215:60ff:fe7a:adc0/128	2620:0:a03:e102:215:60ff:fe7a:adc0	All 128 bits must match. This specifies a single host address.
2001:db8:a03:e102:0:ab4:100::/112	2001:db8:a03:e102:0:ab4:100	The leftmost 112 bits must match. The remaining 16 bits are wildcards.

Configuration Commands

Command Summary for Configuring ACLs

Create an IPv6 ACL *or* Add an ACE to the End of an Existing IPv6 ACL	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# < deny | permit > < ipv6 | esp | ah | sctp | *ipv6-protocol-nbr* > < any | host <*SA* > | *SA*/< *prefix-length* >> < any | host < *DA* > | *DA*/< *prefix-length* >> < tcp | udp > < any | host <*SA* > | *SA*/< *prefix-length* > > [*comparison-operator* < *value* >] < any | host < *DA* > | *DA*/< *prefix-length* >> [*comparison-operator* < *value* >] [established][1] [ack] [fin] [rst] [syn][2] < icmp > < any | host < *SA* > | *SA* /< *prefix-length* >> < any | host < *DA* > | *DA* /< *prefix-length* >> [0 - 255 [0 - 255] | *icmp-message*] [dscp < *precedence* | *codepoint* >] [log][3]	8-47
Insert an ACE by Assigning a Sequence Number	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# < *seq-#* > < deny | permit > *The deny and permit keywords use the options shown above for "Create an IPv6 ACL".*	8-68
Delete an ACE or a Remark by Sequence Number	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# no < *seq-#* > [remark] (**Note:** You can also delete an ACE by entering no < permit | deny > followed by the settings explicitly configured for that ACE.)	8-70
Resequence the ACEs in an ACL	ProCurve(config)# ipv6 access-list resequence < *name-str* > < *starting-#* > < *increment* >.	8-71

[1]TCP only.
[2]TCP flag (control bit) options for destination TCP.
[3]The **log** function is available only for "deny" ACLs, and generates a message only when there is a "deny" match.

— Continued —

Continued from preceding page. —

Action	Command(s)	Page
Enter a Remark	ProCurve(config)# ipv6 access-list < *name-str* > ProCurve(config-ipv6-acl)# remark < *remark-str* >	8-73 8-75
Remove a Remark: – Immediately After Entry – After entry of an ACE	 ProCurve(config-ipv6-acl)# no remark ProCurve(config-ipv6-acl)#no < *seq-#* > remark	
Delete an IPv6 ACL	ProCurve(config)# no ipv6 access-list < *name-str* > vlan	8-65

Command Summary for Enabling, Disabling, and Displaying ACLs

Enable or Disable an IPv6 VACL	ProCurve(config)# [no] vlan < *vid* > ipv6 access-group < *name-str* > vlan	
Enable or Disable a Static Port ACL	ProCurve(config)# [no] interface < *port-list* | trk*x* > ipv6 access-group < *name-str* > in ProCurve(eth-< *port-list* >| trk*x* >)# [no] ipv6 access-group < *name-str* > in	
Displaying ACL Data	ProCurve(config)# show access-list ProCurve(config)# show access-list < *acl-name-str* > [config] ProCurve(config)# show access-list config ProCurve(config)# show access-list ports < *port-list* > ProCurve(config)# show access-list vlan < *vid* > ProCurve(config)# show access-list radius < *port-list* | all > ProCurve(config)# show access-list resources	8-78

Overview

IPv6 ACLs enable filtering on the following:

- Source and destination IPv6 addresses (required), in one of the following options:
  - specific host IPv6
  - subnet or contiguous set of IPv6 addresses
  - any IPv6 address
- choice of any IPv6 protocol
- optional packet-type criteria for ICMP traffic
- optional source and/or destination TCP or UDP port, with a further option for comparison operators

- TCP flag (control bit) options
- filtering for TCP traffic based on whether the subject traffic is initiating a connection ("established" option)
- optional DSCP (IP precedence and ToS) criteria

The switch allows up to 2048 ACLs each for IPv4 and IPv6 (with RADIUS-based ACL resources drawn from the IPv4 allocation). The total is determined from the number of unique identifiers in the configuration. For example, configuring two IPv6 ACLs results in an ACL total of two, even if neither is assigned to an interface. If you then assign a nonexistent IPv6 ACL to an interface, the new total is three, because the switch now has three unique IPv6 ACL names in its configuration.

- For information on determining the current resource availability and usage, refer to the appendix titled "Monitoring Resources" in the *Management and Configuration Guide* for your switch.
- For ACL resource limits, refer to the appendix covering scalability in the latest *Management and Configuration Guide* for your switch.

Commands To Create, Enter, and Configure an ACL

For a match to occur with an ACE, a packet must have the source and destination IPv6 address criteria specified by the ACE, as well as any IPv6 protocol-specific criteria included in the command.

Use the following general steps to create or add to an ACL:

1. Create and/or enter the context of a given ACL.
2. Enter the first ACE in a new ACL or append an ACE to the end of an ACL.

This section describes the commands for performing these steps. For other ACL topics, refer to the following:

Topic	Page
applying or removing an ACL on an interface	8-62
deleting an ACL	8-65
editing an ACL (inserting or removing ACEs from an existing ACL)	8-66
sequence numbering in ACLs	8-66
including remarks in an ACL	8-73
displaying ACL configuration data	8-78

Topic	Page
creating or editing ACLs offline	8-87
enabling ACL “Deny” logging	8-92

Creating an ACL and/or Entering the IPv6 ACL (ipv6-acl) Context.

This command is a prerequisite for entering or editing ACEs in an ACL. (For a summary of the ACL syntax options, refer to "Command Summary for Configuring ACLs" on page 8-45.)

Syntax: ipv6 access-list < *ascii-str* >

*Places the CLI in the IPv6 ACL (***ipv6-acl***) context specified by the < ascii-str > alphanumeric identifier. This enables entry of individual ACEs in the specified ACL. If the ACL does not already exist, this command creates it.*

< ascii-str >: *Specifies an alphanumeric identifier for the ACL. Consists of an alphanumeric string of up to 64 case-sensitive characters. Including spaces in the string requires that you enclose the string in single or double quotes. For example: "***Accounting ACL***". You can also use this command to access an existing ACL. Refer to "General Editing Rules" on page 8-66*

```
ProCurve(config)# ip access-list Sample-List
ProCurve(config-ipv6-acl)#
```

Figure 8-10. Example of Entering the ACL Context

Configure ACEs in an ACL. Configuring ACEs is done after using the **ipv6 access-list <*ascii-str*>** command described on page 8-49 to enter the IPv6 ACL (**ipv6-acl**) context of an ACL. For an IPv6 ACL syntax summary, refer to "Command Summary for Configuring ACLs" on page 8-45.

Syntax: (ipv6 acl context) < deny | permit > < ipv6 | *ipv6-protocol* | *ipv6-protocol-nbr* >
< any | host < *SA* > | *SA/ prefix-length* >
< any | host < *DA* > | *DA/ prefix-length* >
[dscp < *tos-bits* | *precedence*] [log]

Appends an ACE to the end of the list of ACEs in the current ACL. In the default configuration, ACEs are automatically assigned consecutive sequence numbers in increments of 10 and can be renumbered using **resequence** *(page 8-71).*

Note: *To insert a new ACE between two existing ACEs in an ACL, precede* **deny** *or* **permit** *with an appropriate sequence number. (Refer to "Inserting an ACE in an Existing ACL" on page 8-68.)*
For a match to occur, a packet must have the source and destination IPv6 addressing criteria specified in the ACE, as well as:

- *the protocol-specific criteria configured in the ACE, including any optional elements (described later in this section)*
- *any (optional) DSCP settings configured in the ACE*

< deny | permit >

*These keywords are used in the IPv6 (***ipv6-acl***) context to specify whether the ACE denies or permits a packet matching the criteria in the ACE, as described below.*

< ipv6 | *ipv6-protocol* | *ipv6-protocol-nbr* >

Used after **deny** *or* **permit** *to specify the packet protocol type required for a match. An ACL must include one of the following:*

- **ipv6** — *any IPv6 packet.*
- ***ipv6-protocol*** — *any one of the following IPv6 protocol names:*

  esp **ah** **sctp** **icmp***
  tcp* **udp***

  ** For TCP, UDP, and ICMP, additional, optional criteria can be specified, as described on pages 8-55 through 8-59.*

- ***ipv6-protocol-nbr*** — *the protocol number of an IPv6 packet type, such as "8" for Exterior Gateway Protocol or 121 for Simple Message Protocol. (Range: 0 - 255)*

  (For a listing of IPv6 protocol numbers and their corresponding protocol names, refer to the IANA protocol number assignments at **www.iana.com***..)*

< any | host < *SA* > | *SA* < *prefix-length* >>

This is the first instance of IPv6 addressing in an ACE. It follows the protocol specifier and defines the source IPv6 address (SA) a packet must carry for a match with the ACE.

- **any** — *Allows IPv6 packets from any IPv6 SA.*
- **host < *SA* >** — *Specifies only packets having a single address as the SA. Use this criterion when you want to match only the IPv6 packets from a single SA.*
- ***SA* < *prefix-length* >** — *Specifies packets received from one or more contiguous subnets or contiguous addresses within a single subnet. The prefix length is in CIDR format and defines the number of leftmost bits to use in determining a match. (Refer to "Using CIDR Notation To Enter the IPv6 ACL Prefix Length" on page 8-43.) In a given ACE, the SA prefix length defines how many leftmost bits in a packet's SA must exactly match the SA configured in the ACE.*

  Examples of Prefix-Length Applications:

  - 2001:db8:0:e102::10:100/120 matches any IPv6 address in the range of 2001:db8:0:e102::10:<0100 - 01FF>
  - 2001:db8:a0:e102::/64 matches any IPv6 address having a prefix of 2001:db8:a0:e102.
  - FE80::/16 matches any link-local address on an interface.

Note: *For more on how prefix lengths are used in IPv6 ACLs, refer to "How an ACE Uses a Prefix To Screen Packets for SA and DA Matches" on page 8-33.*

< any | host < *DA* > | *DA/prefix*-length >

This is the second instance of addressing in an IPv6 ACE. It follows the first (SA) instance, described earlier in this section, and defines the destination IPv6 address (DA) that a packet must carry to have a match with the ACE.

- **any** — *Allows IPv6 packets to any IPv6 DA.*
- **host** < ***DA*** > — *Specifies only packets having* ***DA*** *as the destination address. Use this criterion when you want to match only the IPv6 packets for a single DA.*
- ***DA/prefix-length*** **—** *Specifies packets intended for one or more contiguous subnets or contiguous addresses within a single subnet. The prefix length is in CIDR format and defines the number of leftmost bits to use in determining a match. (Refer to "Using CIDR Notation To Enter the IPv6 ACL Prefix Length" on page 8-43.) In a given ACE, the DA prefix length defines how many leftmost bits in a packet's DA must exactly match the DA configured in the ACE.*

  Example: *Refer to "Examples of Prefix-Length Applications" in the presiding syntax description.*

[dscp < *codepoint* | *precedence* >]

This option follows the DA to include a DSCP codepoint or precedence as a matching criteria.

codepoint: *Supports these codepoint selection options:*

0 - 63*: Select a specific DSCP codepoint by entering its decimal equivalent. (Refer to table 8-4, "DSCP Codepoints with Decimal Equivalents" on page 8-54*

Assured Forwarding (AF) codepoint matches:

AF	*DSCP Match*	*AF*	*DSCP Match*
af11	001010	af31	011010
af12	001100	af32	011100
af13	001110	af33	011110
af21	010010	af41	100010
af22	010100	af42	100100
af23	010110	af43	100110

default: Matches with the 000000 (default) DSCP.

ef*: Expedited Forwarding (EF; 000000) DSCP match.*

precedence: *Supports selection of a precedence setting in the DSCP.*

Option	*Precedence Bits*	*Name*
cs1	001	priority
cs2	010	immediate
cs3	011	flash
cs4	100	flash-override
cs5	101	critical
cs6	110	internet (for internetwork control)
cs7	111	network (for network control)

Note: *The precedence criteria described in this section are applied in addition to any other selection criteria configured in the same ACE. Also, where* **dscp** *is configured in a given ACE, the* **established** *keyword and the optional TCP control bits cannot be configured.*

[log]

This option can be used after the DA to generate an Event Log message if:

- *The action is* **deny**. *(Not applicable to* **permit** *actions.)*
- *There is a match.*
- *ACL logging is enabled. (Refer to "Enabling ACL Logging on the Switch" on page 8-93.)*

For a given ACE, if **log** is used, it must be the last keyword entered.

Table 8-4. DSCP Codepoints with Decimal Equivalents

DSCP Bits	Decimal	DSCP Bits	Decimal	DSCP Bits	Decimal
000000	0 (default)	010110	22 (3*)	101011	43
000001	1	010111	23	101100	44
000010	2	011000	24	101101	45
000011	3	011001	25	101110	46 (7**)
000100	4	011010	26 (4*)	101111	47
000101	5	011011	27	110000	48
000110	6	011100	28 (4*)	110001	49
000111	7	011101	29	110010	50
001000	8	011110	30 (5*)	110011	51
001001	9	011111	31	110100	52
001010	10 (1*)	100000	32	110101	53
001011	11	100001	33	110110	54
001100	12 (1*)	100010	34 (6*)	110111	55
001101	13	100011	35	111000	56
001110	14 (2*)	100100	36 (6*)	111001	57
001111	15	100101	37	111010	58
010000	16	100110	38 (7*)	111011	59
010001	17	100111	39	111100	60
010010	18 (0 *)	101000	40	111101	61
010011	19	101001	41	111110	62
010100	20 (0 *)	101010	42	111111	63
010101	21				

*Assured Forwarding codepoint and 802.1p precedence.
**Expedited Forwarding codepoint configured by default.

Options for TCP and UDP Traffic in IPv6 ACLs. An ACE designed to permit or deny TCP or UDP traffic can optionally include port number criteria for either the source or destination, or both. Use of TCP criteria also allows the **established** option for controlling TCP connection traffic. (For a summary of the syntax options, refer to "Command Summary for Configuring ACLs" on page 8-45..)

TCP: < deny | permit > tcp
< *SA* > [*comparison-operator* < *tcp-src-port* >]
< *DA* > [*comparison-operator* < *tcp-dest-port* >]
[established]
[ack] [fin] [rst] [syn]

UDP: < deny | permit > udp
< *SA* > [*comparison-operator* < *udp-src-port* >]
< *DA* > [*comparison-operator* < *udp-dest-port* >]

In an IPv6 ACL using either **tcp** *or* **udp** *as the IP packet protocol type, you can optionally apply comparison operators specifying TCP or UDP source and/or destination port numbers or ranges of numbers to further define the criteria for a match. For example:*

```
#deny tcp host fe80::119 eq 23 host fe80::155
 established
#permit tcp host 2001:db8::10.100 host
    2001:db8::15:12 eq telnet
#deny udp 2001:db8::ad5:1f4 host 2001:db8::ad0:ff3
    range 161 162
```

[*comparison-operator* < *tcp/udp-src-port* >]

To specify a TCP or UDP source port number in an ACE, (1) select a comparison operator from the following list and (2) enter the port number or a well-known port name.

Comparison Operators:

- **eq** < ***tcp/udp-port-nbr*** > — *"Equal To"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be equal to* < ***tcp/udp-port-nbr*** >.
- **gt** < ***tcp/udp-port-nbr*** > — *"Greater Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be greater than* < ***tcp/udp-port-nbr*** >.
- **lt** < ***tcp/udp-port-nbr*** > — *"Less Than"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must be less than* < ***tcp/udp-port-nbr*** >.
- **neq** < ***tcp/udp-port-nbr*** > — *"Not Equal"; to have a match with the ACE entry, the TCP or UDP source port number in a packet must not be equal to* ***< tcp/udp-port-nbr >***.
- **range** ***< start-port-nbr > < end-port-nbr >*** — *For a match with the ACE entry, the TCP or UDP source-port number in a packet must be in the range* <***start-port-nbr > < end-port-nbr***>.

Port Number or Well-Known Port Name:

Use the TCP or UDP port number required by your application. The switch also accepts these well-known TCP or UDP port names as an alternative to their port numbers:

- ***TCP***: bgp, dns, ftp, http, imap4, ldap, nntp, pop2, pop3, smtp, ssl, telnet
- ***UDP***: bootpc, bootps, dns, ntp, radius, radius-old, rip, snmp, snmp-trap, tftp

To list the above names, press the **[Shift] [?]** *key combination after entering an operator. For a comprehensive listing of port numbers, visit* **www.iana.org/assignments/port-numbers**.

[*comparison-operator* < *tcp-dest-port* >] [established]
[*comparison-operator* < *udp-dest-port* >]

This option, if used, is entered immediately after the ***< DA >*** *entry. To specify a TCP or UDP port number, (1) select a comparison operator and (2) enter the port number or a well-known port name.*

Comparison Operators and Well-Known Port Names — *These are the same as are used with the TCP/UDP source-port options, and are listed earlier in this command description.*

[established] — *This option applies only where TCP is the configured IPv6 protocol type. It blocks the synchronizing packet associated with establishing a new TCP connection while allowing all other IPv6 traffic for existing connections. For example, a Telnet connect requires TCP traffic to move both ways between a host and the target device. Simply applying a* **deny** *to inbound Telnet traffic on a VLAN would prevent Telnet sessions in either direction because responses to outbound requests would be blocked. However, by using the* **established** *option, inbound Telnet traffic arriving in response to outbound Telnet requests would be permitted, but inbound Telnet traffic trying to establish a new connection would be denied. The* **established** *and* **dscp** *options are mutually exclusive in a given ACE. Configuring* **established** *and any combination of TCP control bits in the same ACE is supported, but* **established** *must precede any TCP control bits configured in the ACE.*

TCP Control Bits. *In a given ACE for filtering TCP traffic you can configure one or more of these options:*

[ack] — *Acknowledgement.*

[fin] — *Sender finished.*

[rst] — *Connection reset.*

[syn] — *TCP control bit: sequence number synchronize.*

For more on using TCP control bits, refer to RFC 793.

Options for Filtering ICMP Traffic. This option allows configuring an ACE to selectively permit some types of ICMP traffic while denying other types. An ACE designed to permit or deny ICMP traffic can optionally include an ICMP type and code value to permit or deny an individual type of ICMP packet while not addressing other ICMP traffic types in the same ACE. As a further option, the ACE can include the name of an ICMP packet type. (For a summary of the syntax options, refer to "Command Summary for Configuring ACLs" on page 8-45.)

Syntax: < deny | permit > icmp < *SA* > < *DA* > [*icmp-type* [*icmp-code*]]
< deny | permit > icmp < *SA* > < *DA* > [*icmp-type-name*]

Using **icmp** *as the packet protocol type, you can optionally specify an individual ICMP packet type or packet type/code pair to further define the criteria for a match. This option, if used, is entered immediately after the destination IP address (DA) entry. The following example shows two ACEs entered in an ACL context:*

```
#permit icmp any any 1 3
#permit icmp any any destination-unreachable
```

[*icmp-type* [*icmp-code*]]

This option identifies an individual ICMP packet type as criteria for permitting or denying that type of ICMP traffic in an ACE.

- **icmp-type —** *This value is in the range of 0 - 255 and corresponds to an ICMP packet type.*
- **icmp-code —** *This value corresponds to an ICMP code for an ICMP packet type. It is optional, and needed only when a particular ICMP subtype is needed as a filtering criterion. (Range: 0 - 255)*

For example, the following ACE specifies "destination unreachable" (ICMP type 1) where "address unreachable" (3; a subtype of "destination unreachable") is the specific code.

```
#permit icmp any any 1 3
```

For more information on ICMP types and codes, visit the Internet Assigned Numbers Authority (IANA) website at **www.iana.com,** *and refer to "Internet Control Message Protocol version 6 (ICMPv6) Type Numbers".*

[*icmp-type-name*]

These name options are an alternative to the **[icmp-type [icmp-code]]** *methodology described above. For more information, visit the IANA website cited above.*

cert-path-advertise	mobile-advertise
cert-path-solicit	mobile-solicit
destination-unreachable	nd-na
echo-reply	nd-ns
echo-request	node-info
home-agent-reply	node-query
home-agent-request	packet-too-big
inv-nd-na	parameter-problem
inv-nd-ns	redirect
mcast-router-advertise	router-advertisement
mcast-router-solicit	router-renum
mcast-router-terminate	router-solicitation
mld-done	time-exceeded
mld-query	ver2-mld-report
mld-report	

Example of an IPv6 ACL Configuration. Suppose that you wanted to implement the following IPv6 traffic policy on a switch connecting two workgroups on the same VLAN to dedicated servers and to a campus intranet (figure 8-11 on page 8-60):

- Permit full IPv6 access for the management station.
- For traffic from the workgroup at 2001:db8::1:20:0/121:
  - Deny Telnet access to server “1” (2001:db8::1:10:3).
  - Deny the workgroup any IPv6 access to server “2” (2001:db8::1:10:4).
- For traffic from the workgroup at 2001:db8::1:30:0/121:
  - Deny Telnet access to server “2” (2001:db8::1:10:4).
  - Deny the workgroup any IPv6 access to server (2001:db8::1:10:3).
- Deny inbound ICMP multicast-router-solicitations from all switches on the VLAN.

- Permit all other IPv6 traffic. (Supersedes the implicit **deny ipv6 any any** at the end of the ACL, which would deny any IPv6 traffic not filtered by the configured ACEs in the ACL.)

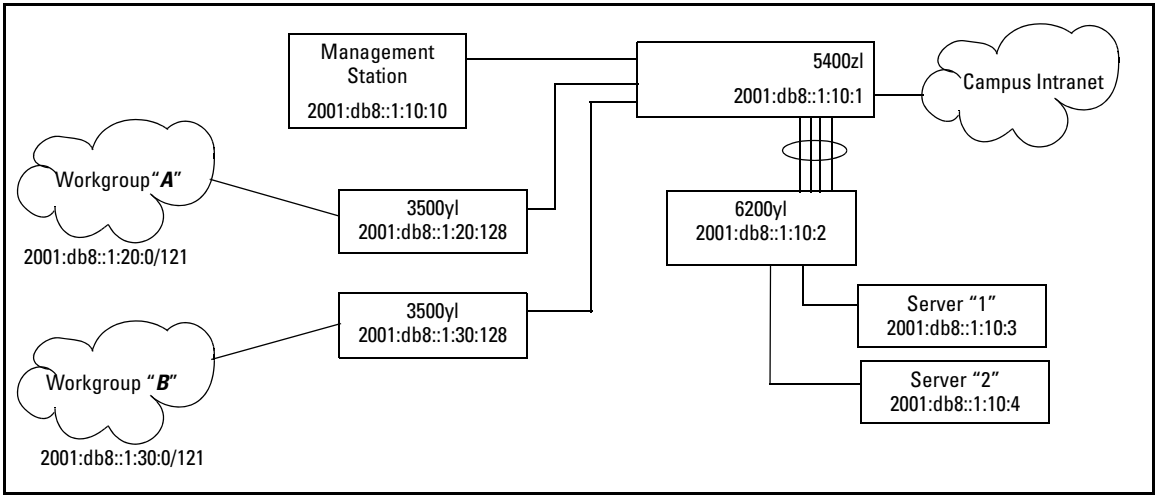

Figure 8-11. Example of Controlling Workgroup Access to Servers

Continuing the example, you would use the following commands to configure the ACL:

```
ProCurve(config)# ipv6 access-list Test-01
ProCurve(config-ipv6-acl)#
ProCurve(config-ipv6-acl)#permit ipv6 host 2001:db8::1:10:10 any
ProCurve(config-ipv6-acl)#deny tcp 2001:db8::1:20:0/121 host 2001:db8::1:10:3 eq
telnet log
ProCurve(config-ipv6-acl)#deny  ipv6  2001:db8::1:20:0/121  host  2001:db8::1:10:4
log
ProCurve(config-ipv6-acl)#deny tcp 2001:db8::1:30:0/121 host 2001:db8::1:10:4 eq
telnet log
ProCurve(config-ipv6-acl)#deny ipv6 2001:db8::1:30:0/121 host 2001:db8::1:10:3
ProCurve(config-ipv6-acl)#deny icmp any any router-solicitation
ProCurve(config-ipv6-acl)#permit ipv6 any any
ProCurve(config-ipv6-acl)#exit
```

Figure 8-12. Commands To Configure an IPv6 ACL To Control Access to the Servers in Figure 8-11

The configuration of the example in the switch appears as follows:

```
Port-1(config)# show access-list config

ipv6 access-list "Test-01"
     10 permit ipv6 2001:db8::1:10:10/128 ::/0
     20 deny tcp 2001:db8::1:20:0/121 2001:db8::1:10:3/128 eq 23 log
     30 deny ipv6 2001:db8::1:20:0/121 2001:db8::1:10:4/128 log
     40 deny tcp 2001:db8::1:30:0/121 2001:db8::1:10:4/128 eq 23 log
     50 deny ipv6 2001:db8::1:30:0/121 2001:db8::1:10:3/128
     60 deny icmp ::/0 ::/0 133
     70 permit ipv6 ::/0 ::/0
   exit
```

Figure 8-13. CLI Listing of the ACL Entered by the Commands in Figure 8-12

Adding or Removing an ACL Assignment On an Interface

Filtering Switched IPv6 Traffic Inbound on a VLAN

For a given VLAN interface, you can assign an ACL as a VACL to filter switched IPv6 traffic entering the switch on that VLAN. You can also use the same ACL for assignment to multiple VLANs. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 8-31.

Syntax: [no] vlan < *vid* > ipv6 access-group < *identifier* > vlan

Assigns an ACL as a VACL to a VLAN to filter switched IPv6 traffic entering the switch on that VLAN. You can use either the global configuration level or the VLAN context level to assign or remove a VACL.

< *vid* >: *VLAN Identification Number.*

< *identifier* >: *The alphanumeric name by which the ACL can be accessed. An identifier can have up to 64 characters.*

The **no** *form of the command removes the ACL assignment from the interface.*

Note: *The switch allows you to assign an "empty" ACL identifier to a VLAN. In this case, if you later populate the ACL with ACEs, the new ACEs automatically become active on the assigned VLAN as they are created. Also, if you delete an assigned ACL from the switch without also using the "***no***" form of this command to remove the assignment to a VLAN, the ACL assignment remains as an "empty" ACL. For more on "empty" ACLs, refer to the notes under "Deleting an ACL" on page 8-65.*

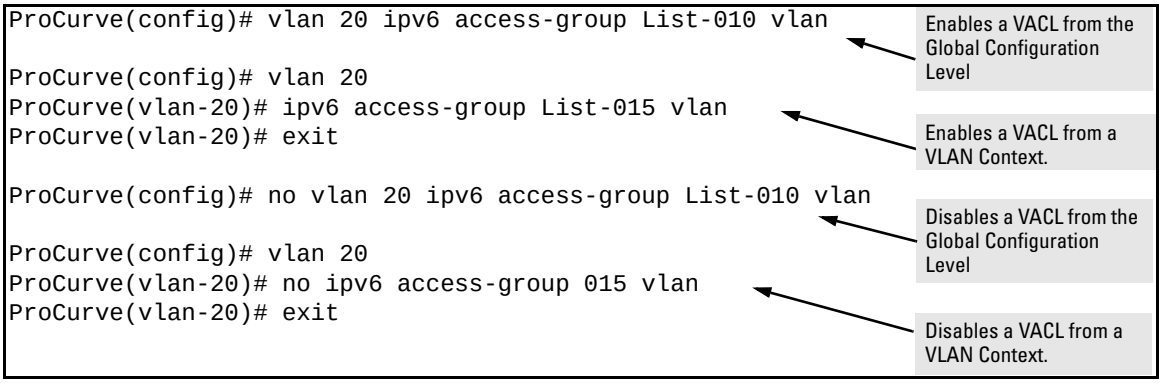
ProCurve(config)# vlan 20 ipv6 access-group List-010 vlan
ProCurve(config)# vlan 20
ProCurve(vlan-20)# ipv6 access-group List-015 vlan
ProCurve(vlan-20)# exit
ProCurve(config)# no vlan 20 ipv6 access-group List-010 vlan
ProCurve(config)# vlan 20
ProCurve(vlan-20)# no ipv6 access-group 015 vlan
ProCurve(vlan-20)# exit
Enables a VACL from the Global Configuration Level
Enables a VACL from a VLAN Context.
Disables a VACL from the Global Configuration Level
Disables a VACL from a VLAN Context.

Figure 8-14. Methods for Enabling and Disabling VACLs

Filtering Inbound IPv6 Traffic Per Port and Trunk

For a given port, port list, or static port trunk, you can assign an ACL as a static port ACL to filter switched IPv6 traffic entering the switch on that interface. You can also use the same ACL for assignment to multiple interfaces. For limits and operating rules, refer to "ACL Configuration and Operating Rules" on page 8-31.

Syntax: [no] interface < *port-list* | Trk*x* > ipv6 access-group < *identifier* > in

Assigns an ACL as a static port ACL to a port, port list, or static trunk to filter switched IPv6 traffic entering the switch on that interface. You can use either the global configuration level or the interface context level to assign or remove a static port ACL.

< *identifier*>: *The alphanumeric name by which the ACL can be accessed. An identifier can have up to 64 characters.*

< *port-list* | trk*x* >: *The port, trunk, or list of ports and/or trunks on which to assign or remove the specified ACL.*

Note: *The switch allows you to assign an "empty" ACL identifier to an interface. In this case, if you later populate the empty ACL with one or more ACEs, it automatically becomes active on the assigned interface(s). Also, if you delete an assigned ACL from the running config file without also using the "***no***" form of this command to remove the assignment to an interface, the ACL assignment remains and will automatically activate any new ACL you create with the same identifier. Refer to "Empty ACL" on page 8-10.*

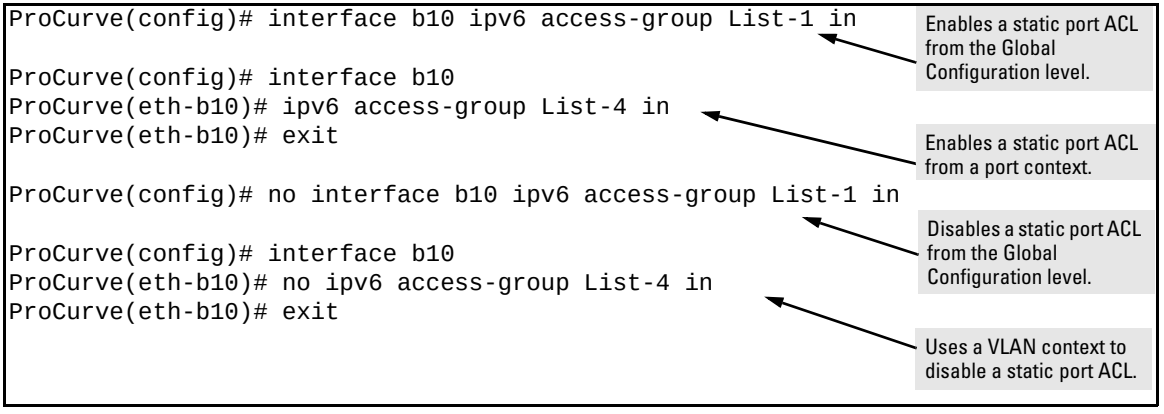
ProCurve(config)# interface b10 ipv6 access-group List-1 in
ProCurve(config)# interface b10
ProCurve(eth-b10)# ipv6 access-group List-4 in
ProCurve(eth-b10)# exit
ProCurve(config)# no interface b10 ipv6 access-group List-1 in
ProCurve(config)# interface b10
ProCurve(eth-b10)# no ipv6 access-group List-4 in
ProCurve(eth-b10)# exit
Enables a static port ACL from the Global Configuration level.
Enables a static port ACL from a port context.
Disables a static port ACL from the Global Configuration level.
Uses a VLAN context to disable a static port ACL.

Figure 8-15. Methods for Enabling and Disabling ACLs

Deleting an ACL

Syntax: no ipv6 access-list < *identifier* >

Used in the global config context to remove the specified IPv6 ACL from the switch's running-config file.

< identifier >: *The alphanumeric name by which the ACL can be accessed.*

Notes: *If an ACL name is assigned to an interface before the ACL itself has actually been created, then the switch creates an "empty" version of the ACL in the running configuration and assigns the empty ACL to the interface. Subsequently populating the empty ACL with explicit ACEs causes the switch to automatically activate the ACEs as they are created and to implement the implicit deny at the end of the ACL.*

Deleting an ACL from the running configuration while the ACL is currently assigned on an interface results in an "empty" version of the ACL in the running configuration and on the interface. Subsequently removing the ACL from the interface also removes the empty ACL from the running configuration.

If you need to remove an ACL identifier assignment on an interface, refer to "Adding or Removing an ACL Assignment On an Interface" on page 8-62

Editing an Existing ACL

The CLI provides the capability for editing in the switch by using sequence numbers to insert or delete individual ACEs. An offline method is also available. This section describes using the CLI for editing ACLs. To use the offline method for editing ACLs, refer to "Creating or Editing ACLs Offline" on page 8-87.

General Editing Rules

You can use the CLI to delete individual ACEs from anywhere in an ACL, append new ACEs to the end of an ACL, and insert new ACEs anywhere within an ACL.

- When you enter a new ACE in an ACL without specifying a sequence number, the switch inserts the ACE as the last entry in the ACL.
- When you enter a new ACE in an ACL and include a sequence number, the switch inserts the ACE according to the position of the sequence number in the current list of ACEs.
- You can delete an ACE by using the **ipv6 access-list < *identifier* >** command to enter the ACL's context, and then **no < *seq-#* >** (page 8-70).
- Deleting the last ACE from an ACL leaves the ACL in the configuration as an "empty" ACL placeholder that cannot perform any filtering tasks. (In any ACL the Implicit Deny does not apply unless the ACL includes at least one explicit ACE.) (Refer to the Notes on the preceding page and to "Empty ACL" on page 8-10.)

Sequence Numbering in ACLs

The ACEs in any ACL are sequentially numbered. In the default state, the sequence number of the first ACE in a list is "10" and subsequent ACEs are numbered in increments of 10. For example, the following **show run** output shows an ACL named "My-list" using the default numbering scheme:

```
ipv6 access-list "My-list"
     10 permit ipv6 2001:db8:0:5ad::25/128 ::/0
     20 permit ipv6 2001:db8:0:5ad::111/128 ::/0
     30 permit icmp 2001:db8:0:5ad::115/128 ::/0 135
     40 deny ipv6 2001:db8:0:5ad::/64 ::/0
   exit
```

Figure 8-16. Example of the Default Sequential Numbering for ACEs

An ACE can be appended to the end of the ACL by using **ipv6 access-list** from the global configuration prompt or by entering the ACL context:

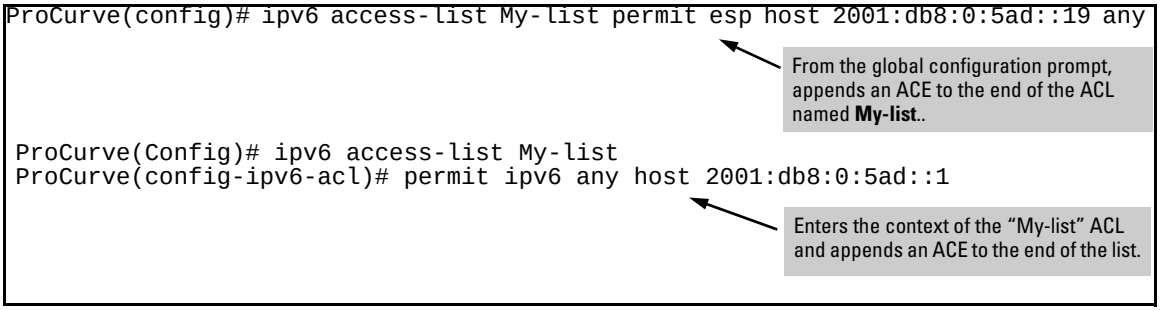
ProCurve(config)# ipv6 access-list My-list permit esp host 2001:db8:0:5ad::19 any
From the global configuration prompt, appends an ACE to the end of the ACL named **My-list**..
ProCurve(Config)# ipv6 access-list My-list
ProCurve(config-ipv6-acl)# permit ipv6 any host 2001:db8:0:5ad::1
Enters the context of the "My-list" ACL and appends an ACE to the end of the list.

Figure 8-17. Examples of Ways to Append a New ACE to the end of an ACL

To continue from figure 8-17 and append a final ACE to the end of the ACL :

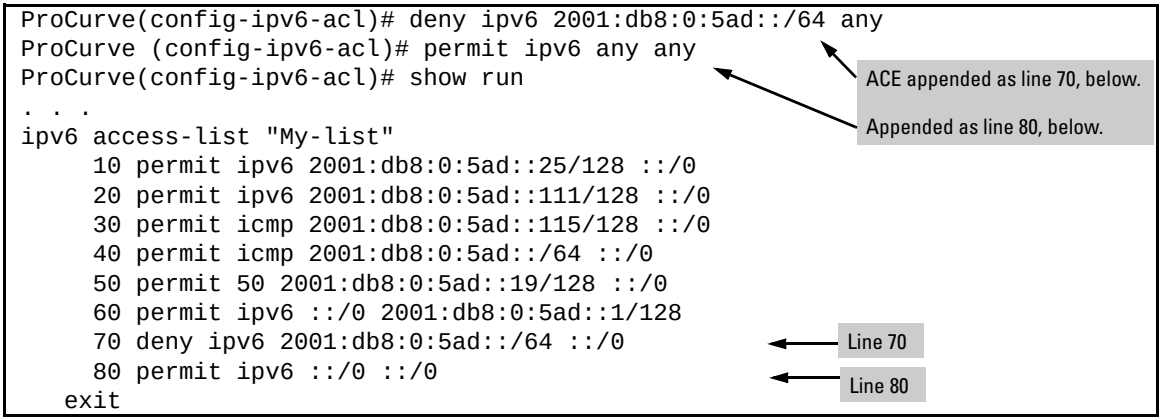
ProCurve(config-ipv6-acl)# deny ipv6 2001:db8:0:5ad::/64 any
ProCurve (config-ipv6-acl)# permit ipv6 any any
ProCurve(config-ipv6-acl)# show run
. . .
ipv6 access-list "My-list"
10 permit ipv6 2001:db8:0:5ad::25/128 ::/0
20 permit ipv6 2001:db8:0:5ad::111/128 ::/0
30 permit icmp 2001:db8:0:5ad::115/128 ::/0
40 permit icmp 2001:db8:0:5ad::/64 ::/0
50 permit 50 2001:db8:0:5ad::19/128 ::/0
60 permit ipv6 ::/0 2001:db8:0:5ad::1/128
70 deny ipv6 2001:db8:0:5ad::/64 ::/0
80 permit ipv6 ::/0 ::/0
exit
ACE appended as line 70, below.
Appended as line 80, below.
Line 70
Line 80

Figure 8-18. Example of Appending an ACE to an Existing List

Inserting an ACE in an Existing ACL

This action uses a sequence number to specify where to insert a new ACE into an existing sequence of ACEs in an ACL.

Syntax: <1-2147483647> < permit | deny > < *ipv6-ACE-criteria* >

Used in the context of a given ACL, this command inserts an ACE into the ACL.

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

< *ipv6-ACE-criteria* >: *The various traffic selection options described earlier in this chapter.*

Note: *Entering an ACE that would result in an out-of-range sequence number is not allowed. Use the resequence command to free up ACE numbering availability in the ACL. Refer to "Resequencing the ACEs in an IPv6 ACL" on page 8-71.*

(For details on these options, refer to "Command Summary for Configuring ACLs" on page 8-7.)

Examples of Inserting a New ACE in an Existing ACL. From the global configuration context, insert a new ACE with a sequence number of 45 between the ACEs numbered 40 and 50 in figure 8-18 .

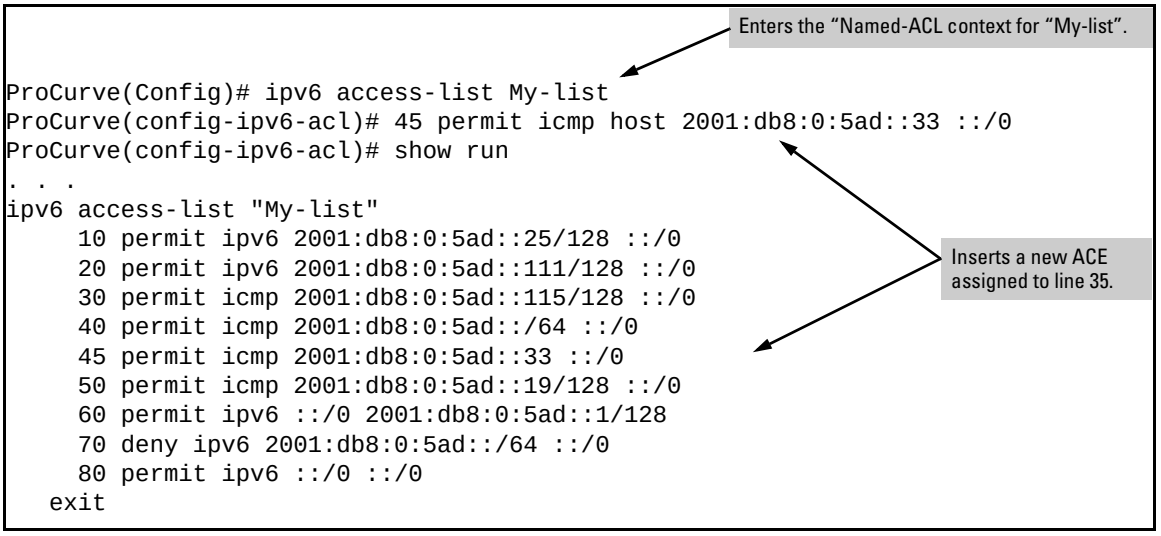
Enters the "Named-ACL context for "My-list".
ProCurve(Config)# ipv6 access-list My-list
ProCurve(config-ipv6-acl)# 45 permit icmp host 2001:db8:0:5ad::33 ::/0
ProCurve(config-ipv6-acl)# show run
. . .
ipv6 access-list "My-list"
10 permit ipv6 2001:db8:0:5ad::25/128 ::/0
20 permit ipv6 2001:db8:0:5ad::111/128 ::/0
30 permit icmp 2001:db8:0:5ad::115/128 ::/0
40 permit icmp 2001:db8:0:5ad::/64 ::/0
45 permit icmp 2001:db8:0:5ad::33 ::/0
50 permit icmp 2001:db8:0:5ad::19/128 ::/0
60 permit ipv6 ::/0 2001:db8:0:5ad::1/128
70 deny ipv6 2001:db8:0:5ad::/64 ::/0
80 permit ipv6 ::/0 ::/0
exit
Inserts a new ACE assigned to line 35.

Figure 8-19. Example of Inserting an ACE in an Existing ACL

From within the context of an IPv6 ACL named "List-01", insert a new ACE between two existing ACEs. In this example, the first command creates a new ACL and enters the ACL context. The next two ACEs entered become lines 10 and 20 in the list. The third ACE entered is inserted between lines 10 and 20 by using the sequence command with a sequence number of 11.

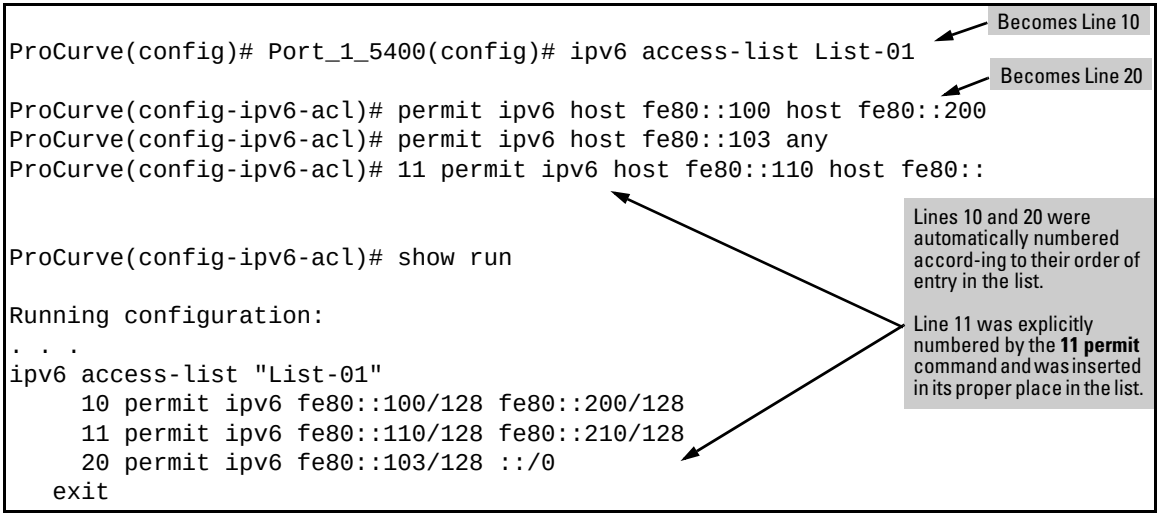
Becomes Line 10
ProCurve(config)# Port_1_5400(config)# ipv6 access-list List-01
Becomes Line 20
ProCurve(config-ipv6-acl)# permit ipv6 host fe80::100 host fe80::200
ProCurve(config-ipv6-acl)# permit ipv6 host fe80::103 any
ProCurve(config-ipv6-acl)# 11 permit ipv6 host fe80::110 host fe80::
ProCurve(config-ipv6-acl)# show run
Running configuration:
. . .
ipv6 access-list "List-01"
10 permit ipv6 fe80::100/128 fe80::200/128
11 permit ipv6 fe80::110/128 fe80::210/128
20 permit ipv6 fe80::103/128 ::/0
exit
Lines 10 and 20 were automatically numbered accord-ing to their order of entry in the list.
Line 11 was explicitly numbered by the **11 permit** command and was inserted in its proper place in the list.

Figure 8-20. Example of Inserting an ACE into an Existing Sequence

Deleting an ACE from an Existing ACL

This action provides the option of using either the sequence number of an ACE or the syntax of the ACE to delete the ACE from an ACL.

Syntax: no <1-2147483647>

no < permit | deny > < *ipv6-ACE-criteria* >

Both command options require entering the configuration context of the ACL containing the ACE you want to delete.

The first command option deletes the ACE assigned to the specified sequence number. The second command option deletes the ACE having the syntax specified by ***< ipv6-ACE-criteria >****.*

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

< ipv6-ACE-criteria >: *The traffic selection options included in the ACE. To use this method to delete an ACE, the criteria specified in the command must match the criteria specified in the actual ACE you want to delete.*

(For details on these options, refer to "Command Summary for Configuring ACLs" on page 8-7.)

1. To find the sequence number of the ACE you want to delete, use **show access-list < *identifier* >** or **show access-list config** to view the ACL.
2. Use **ipv6 access-list < *identifier* > config** to enter the IPv6 ACL (**config-ipv6-acl**) context of the specified ACE.
3. In the IPv6 ACL (**config-ipv6-acl**) context, type **no** and enter the sequence number of the ACE you want to delete.

Figure 8-21 illustrates the process for deleting an ACE from a list:

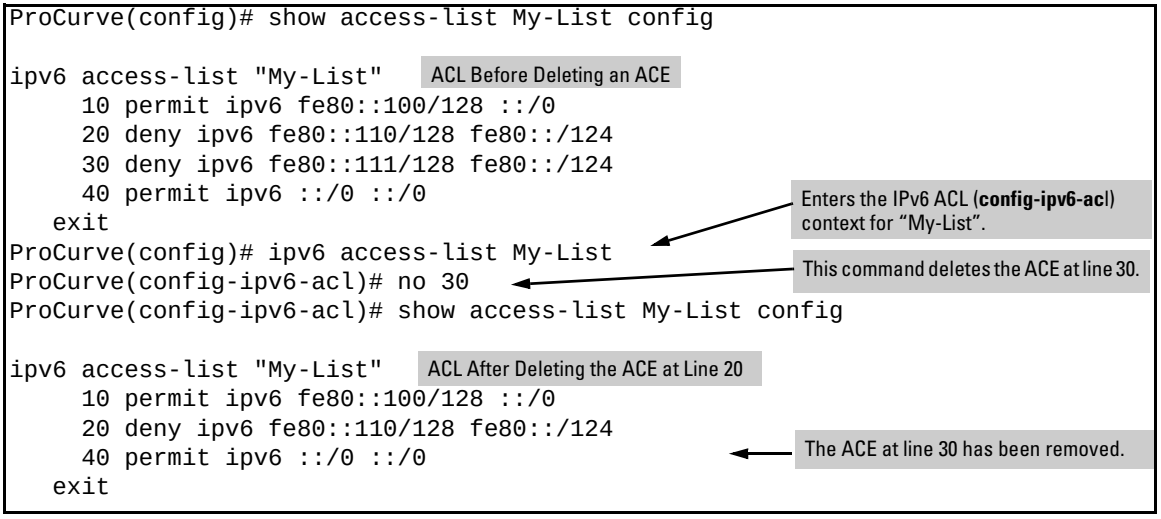
ProCurve(config)# show access-list My-List config
ipv6 access-list "My-List"
ACL Before Deleting an ACE
10 permit ipv6 fe80::100/128 ::/0
20 deny ipv6 fe80::110/128 fe80::/124
30 deny ipv6 fe80::111/128 fe80::/124
40 permit ipv6 ::/0 ::/0
exit
ProCurve(config)# ipv6 access-list My-List
Enters the IPv6 ACL (config-ipv6-acl) context for "My-List".
ProCurve(config-ipv6-acl)# no 30
This command deletes the ACE at line 30.
ProCurve(config-ipv6-acl)# show access-list My-List config
ipv6 access-list "My-List"
ACL After Deleting the ACE at Line 20
10 permit ipv6 fe80::100/128 ::/0
20 deny ipv6 fe80::110/128 fe80::/124
40 permit ipv6 ::/0 ::/0
The ACE at line 30 has been removed.
exit

Figure 8-21. Example of Deleting an ACE from An IPv6 ACL

Resequencing the ACEs in an IPv6 ACL

This action reconfigures the starting sequence number for ACEs in an IPv6 ACL, and resets the numeric interval between sequence numbers for ACEs configured in the ACL.

Syntax: ipv6 access-list resequence < *identifier* > < *starting-seq-#* > < *interval* >

Resets the sequence numbers for all ACEs in the ACL.
< *starting-seq-#* > : *Specifies the sequence number for the first ACE in the list. (Default: 10; Range: 1 - 2147483647)*
< *interval* > : *Specifies the interval between consecutive sequence numbers for the ACEs in the list. (Default: 10; Range: 1 - 2147483647)*

1. To view the current sequence numbering in an ACE, use **show access-list config** or **show access-list < *identifier* > config**.
2. Use the command syntax (above) to change the sequence numbering.

This example resequences the "My-List" ACL at the bottom of figure 8-21 so that the list begins with line 100 and uses a sequence interval of 100.

```
ProCurve(config)# show access-list My-List config

ipv6 access-list "My-List"
     10 permit ipv6 fe80::100/128 ::/0
     20 deny ipv6 fe80::110/128 fe80::/124
     40 permit ipv6 ::/0 ::/0
   exit

ProCurve(config)# ipv6 access-list resequence My-List 100 100
ProCurve(config)# show access-list config

ipv6 access-list "My-List"
     100 permit ipv6 fe80::100/128 ::/0
     200 deny ipv6 fe80::110/128 fe80::/124
     300 permit ipv6 ::/0 ::/0
   exit
```

Figure 8-22. Example of Viewing and Resequencing an ACL

Attaching a Remark to an ACE

A remark is numbered in the same way as an ACE, and uses the same sequence number as the ACE to which it refers. This operation requires that the remark for a given ACE be entered prior to entering the ACE itself.

Syntax: remark < *remark-str* >
< 1-2147483647 > remark < *remark-str* >
no < *seq-#* > remark

These commands are used in the ACL context to enter a comment related to an adjacent ACE. To associate a remark with a specific ACE, do one of the following:

- *Enter the remark first (without a sequence number) and immediately follow it with the ACE (also without a sequence number). The remark and the following ACE will have the same (automatically generated) sequence number.*
- *Enter the ACE with or without a sequence number, then use* **<1-2147483647> remark < *remark-str* >** *to enter the remark, where a number in the range of* **<1-2147483647>** *matches the sequence number of the related ACE. This method is useful when you want to enter a remark at some time after you have entered the related ACE.*

< *remark-str* >: *The text of the remark. If spaces are included in the remark, then the remark string must be delimited by either single quotes or double quotes. For example:*

```
remark Permits_Telnet_from_2001:db8:0:1ab_subnet
remark “Permits Telnet from 2001:db8:0:1ab subnet”
remark ‘Permits Telnet from 2001:db8:0:1ab subnet’
```

<1-2147483647>: *The range of valid sequence numbers for an ACL.*

For example, if the sequence number of the last ACE entered is “30” and sequence numbering is set to the (default) interval of 10, then entering a remark and another ACE without specifying any sequence numbers results in a sequence number of “40” for both the remark and the ACE that follows it.

The **no** *form of the command deletes the indicated remark, but does not affect the related ACE.*

Appending Remarks and Related ACEs to the End of an ACL. To include a remark for an ACE that will be appended to the end of the current ACL, enter the remark first, then enter the related ACE. This results in the remark and the subsequent ACE having the same sequence number. For example, to append an ACE with an associated remark to the end of an ACL named “List-100”, you would enter remarks from the CLI context for the desired ACL:

```
ProCurve(config)# ipv6 access-list List-100
ProCurve(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:17 eq telnet any
ProCurve(config-ipv6-acl)# permit tcp host 2001:db8:0:b::100:23 eq telnet any
ProCurve(config-ipv6-acl)# remark “BLOCKS UNAUTH TELNET TRAFFIC FROM SUBNET B”
ProCurve(config-ipv6-acll)# deny tcp 2001:db8:0:a::/64 eq telnet any
ProCurve(config-ipv6-acl)# show access-list List-100 config

ipv6 access-list "List-100"
     10 remark "TEXT"
     10 permit tcp 2001:db8:0:b::100:17/128 eq 23 ::/0
     20 permit tcp 2001:db8:0:b::100:23/128 eq 23 ::/0
     30 remark "BLOCKS UNAUTH TELNET TRAFFIC FROM SUBNET B"
     30 deny tcp 2001:db8:0:b::/64 eq 23 ::/0
   exit
ProCurve(config-ipv6-acl)#
```

The remark is assigned the same number as the immediately following ACE (“30” in this example) is assigned when it is automatically appended to the end of the list. This operation applies where new remarks and ACEs are appended to the end of the ACL and are automatically assigned a sequence number.

Figure 8-23. Example of Appending a Remark and Its Related ACE to the End of an ACL

Inserting Remarks and Related ACEs Within an Existing List. To insert an ACE with a remark within an ACL by specifying a sequence number, insert the numbered remark first, then, using the same sequence number, insert the ACE. For example:

```
ProCurve(config-ipv6-acl)# 15 remark "PERMIT HTTP; STATION 23; SUBNET 1D"
ProCurve(config-ipv6-acl)#  15  permit  tcp  host  2001:db8:0:1d::23  eq  80
2001:db8:0:2f::/64


ProCurve(config-ipv6-acl)# show access config
. . .

ipv6 access-list "List-105"
     10 permit tcp 2001:db8:0:1f::/64 eq 80 2001:db8:0:2f::/64
     15 remark "PERMIT HTTP; STATION 23; SUBNET 1D"
     15 permit tcp 2001:db8:0:1d::23/128 eq 80 2001:db8:0:2f::/64
     20 deny tcp 2001:db8:0:1d::/64 eq 80 2001:db8:0:2f::/64
   exit
. . .
```

The above two commands insert a remark with its corresponding ACE (same sequence number) between two previously configured ACEs.

Figure 8-24. Example of Inserting a Remark and an ACE Within an Existing ACL

Inserting a Remark for an ACE that Already Exists in an ACL. If an ACE already exists in a given ACL, you can insert a remark for that ACE by simply configuring the remark to have the same sequence number as the ACE.

Replacing an Existing Remark. To replace an existing remark in a given ACL:

1. Use **ipv6 access-list < *identifier* >** to enter the desired ACL context.
2. Configure the replacement remark with the same sequence number as the remark you want to replace. This step overwrites the former remark text with the new remark text.

For example, to change the text of the remark at line 15 in figure 8-24 to "PERMIT HTTP FROM ONE STATION", you would use the following command:

```
ProCurve(config): ipv6 access-list List-105
ProCurve(config-ipv6-acl): 15 remark "PERMIT HTTP FROM ONE STATION"
```

Removing a Remark from an Existing ACE. If you want to remove a remark, but want to retain the ACE, do the following:

1. Use **ipv6 access-list < *identifier* >** to enter the desired ACL context.
2. Use **no <1-2147483647> remark**.

Using the **no <1-2147483647>** command without the remark keyword deletes both the remark and the ACE to which it is attached.

Operating Notes for Remarks

- An "orphan" remark is a remark that does not have an ACE counterpart with the same sequence number. The **resequence** command renumbers an orphan remark as a sequential, standalone entry without a permit or deny ACE counterpart.

```
ipv6 access-list "XYZ"
     10 remark "Permits HTTP"
     10 permit tcp 2001:db8::2:1/120 eq 80 ::/0
     12 remark "Denies HTTP from subnet 1."
     18 remark "Denies pop3 from 1:157."
     18 deny tcp 2001:db8::1:157/128 eq 110 ::/0 log
     50 permit ipv6 ::/0 ::/0
  exit
ProCurve# ipv6 access-list resequence XYZ 100 10
ProCurve# show access-list XYZ config
ipv6 access-list "XYZ"
     100 remark "Permits HTTP"
     100 permit tcp 2001:db8::2:1/120 eq 80 ::/0
     110 remark "Denies HTTP from subnet 1."
     120 remark "Denies pop3 from 1:157."
     120 deny tcp 2001:db8::1:157/128 eq 110 ::/0 log
     130 permit ipv6 ::/0 ::/0
  exit
```

- Entering either an unnumbered remark followed by a manually numbered ACE (using **<1-2147483647>**), or the reverse (an unnumbered ACE followed by a manually numbered remark) can result in an "orphan" remark.

- Configuring two remarks without including either sequence numbers or an intervening, unnumbered ACE results in the second remark overwriting the first.

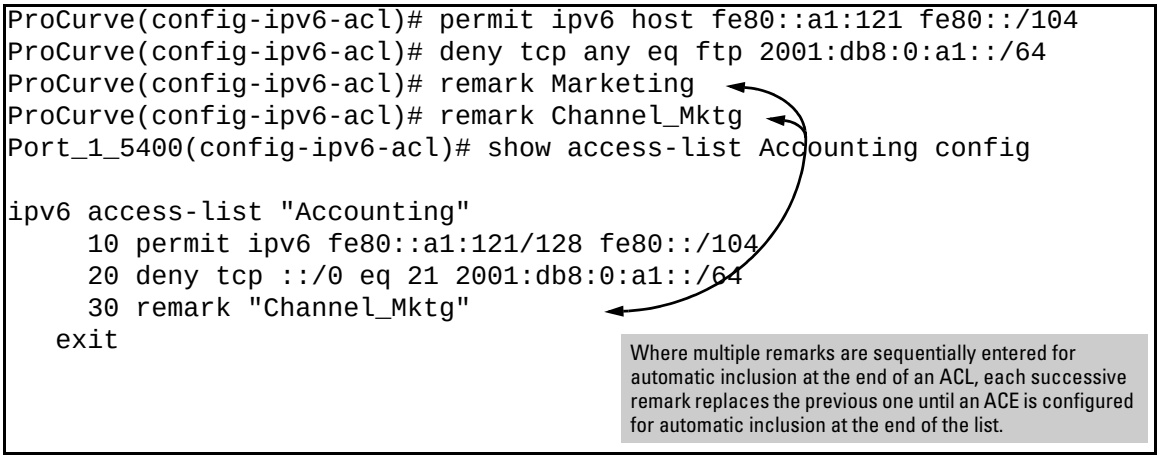
ProCurve(config-ipv6-acl)# permit ipv6 host fe80::a1:121 fe80::/104
ProCurve(config-ipv6-acl)# deny tcp any eq ftp 2001:db8:0:a1::/64
ProCurve(config-ipv6-acl)# remark Marketing
ProCurve(config-ipv6-acl)# remark Channel_Mktg
Port_1_5400(config-ipv6-acl)# show access-list Accounting config
ipv6 access-list "Accounting"
10 permit ipv6 fe80::a1:121/128 fe80::/104
20 deny tcp ::/0 eq 21 2001:db8:0:a1::/64
30 remark "Channel_Mktg"
exit
Where multiple remarks are sequentially entered for automatic inclusion at the end of an ACL, each successive remark replaces the previous one until an ACE is configured for automatic inclusion at the end of the list.

Figure 8-25. Example of Overwriting One Remark with Another

Displaying ACL Configuration Data

The **show** commands in this section apply to both IPv6 and IPv4 ACLs. For information on IPv4 ACL operation, refer to the chapter titled "IPv4 Access Control Lists" in the *Access Security Guide* for your switch.

ACL Commands	Function	Page
show access-list	Displays a brief listing of all IPv4 and IPv6 ACLs on the switch.	8-79
show access-list config	Display the type, identifier, and content of all IPv4 and IPv6 ACLs configured in the switch.	8-80
show access-list vlan < *vid* >	List the name and type for each IPv4 and IPv6 ACL application assigned to a particular VLAN on the switch.	8-81
show access-list ports < all | *port-list* >	Lists the IPv4 and IPv6 ACL static port assignments for either all ports and trunks, or for the specified ports and/or trunks.	8-82
show access-list < *identifier* > [config]	Display detailed content information for a specific IPv4 or IPv6 ACL. Using the config option displays the ACL in a list format similar to that used to display an ACL in the **show running-config** output.	8-83
show access-list resources	Displays the currently available per-slot resource availability. Refer to the appendix titled "Monitoring Resources" in the current *Management and Configuration Guide* for your switch.	
show access-list radius < all | *port-list* >	Lists the IPv4 and IPv6 RADIUS ACLs currently assigned for either all ports and trunks, or for the specified ports and/or trunks. For more on this topic, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the Access Security Guide for your switch.	
show port-access web-based clients < *port-list* > detailed show port-access mac-based clients < *port-list* > detailed show port-access authenticator clients < *port-list* > detailed	For ports in the < port-list > shows the details of the RADIUS-assigned features, including the ACE matches in RADIUS-assigned ACLs configured with the cnt (counter) option. For more on this topic, refer to the chapter titled "Configuring RADIUS Server Support for Switch Services" in the Access Security Guide for your switch.	—
show config show running	**show config** includes configured ACLs and assignments existing in the startup-config file. **show running** includes configured ACLs and assignments existing in the running-config file.	

Display an ACL Summary

This command lists the configured IPv4 and IPv6 ACLs, regardless of whether they are assigned to any interfaces.

Syntax: show access-list

List a summary table of the name, type, and application status of all ACLs (IPv4 and IPv6) configured on the switch.

For example:

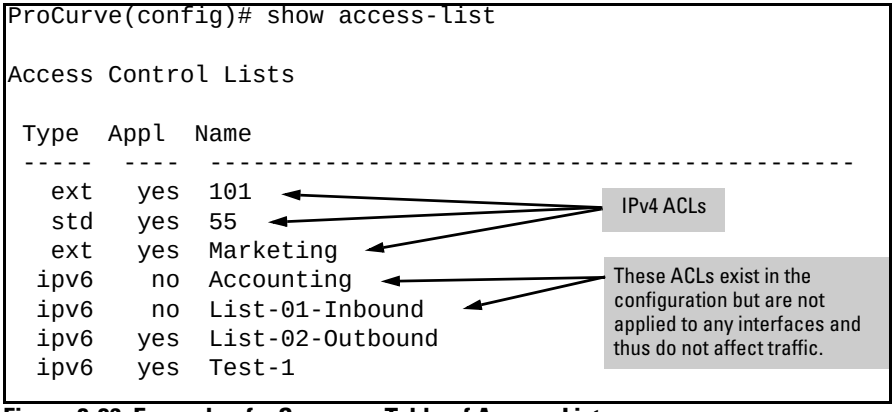
ProCurve(config)# show access-list
Access Control Lists
Type Appl Name
ext yes 101
std yes 55
ext yes Marketing
ipv6 no Accounting
ipv6 no List-01-Inbound
ipv6 yes List-02-Outbound
ipv6 yes Test-1
IPv4 ACLs
These ACLs exist in the configuration but are not applied to any interfaces and thus do not affect traffic.

Figure 8-26. Example of a Summary Table of Access Lists

Term	Meaning
Type	Shows whether the listed ACL is an IPv6 (**ipv6**) ACL or one of two IPv4 ACL types: **std** (Standard; source-address only) or **ext** (Extended; protocol, source, and destination data).
Appl	Shows whether the listed ACL has been applied to an interface (**yes**/**no**).
Name	Shows the identifier assigned toeach ACL configured in the switch.

Display the Content of All ACLs on the Switch

This command lists the configuration details for every IPv4 and IPv6 ACL in the running-config file, regardless of whether any are actually assigned to filter traffic on specific interfaces.

Syntax: show access-list config

List the configured syntax for all IPv4 and IPv6 ACLs currently configured on the switch.

Note

Notice that you can use the output from this command for input to an offline text file in which you can edit, add, or delete ACL commands. Refer to "Creating or Editing ACLs Offline" on page 8-87.

This information also appears in the **show running** output. If you executed **write memory** after configuring an ACL, it appears in the **show config** output.

Figure 8-27 shows the ACLs on a switch configured with two IPv6 ACLs named "Accounting" and "List-01-Inbound", and one extended IPv4 ACL named "101":

```
ProCurve(config)# show access-list config

ip access-list extended "101"
   10 permit tcp 10.30.133.27 0.0.0.0 0.0.0.0 255.255.255.255
   20 permit tcp 10.30.155.101 0.0.0.0 0.0.0.0 255.255.255.255
   30 deny ip 10.30.133.1 0.0.0.0 0.0.0.0 255.255.255.255 log
   40 deny ip 10.30.155.1 0.0.0.255 0.0.0.0 255.255.255.255
   exit
ipv6 access-list "Accounting"
     10 permit tcp 2001:db8:0:1af::10:14/128 ::/0 eq 23
     20 permit tcp 2001:db8:0:1af::10:23/128 ::/0 eq 23
     30 deny tcp 2001:db8:0:1af::10/116 ::/0 log
     40 permit ipv6 2001:db8:0:1af::10/116 ::/0
     50 deny ipv6 ::/0 ::/0 log
   exit
ipv6 access-list "List-01-Inbound"
     10 permit icmp fe80::10:60/128 ::/0 dscp 38
     20 permit icmp fe80::10:77/128 ::/0 dscp 38
     30 permit icmp fe80::10:83/128 ::/0 dscp 38
     40 deny icmp ::/0 ::/0 dscp 38
     50 permit ipv6 fe80::10/112 ::/0
     60 deny ipv6 fe80::/64 ::/0
   exit
```

Figure 8-27. Example of an ACL Configured Syntax Listing

Display the IPv4 and IPv6 VACL Assignments for a VLAN

This command lists the identifiers and type(s) of VACLs currently assigned to a particular VLAN in the running-config file. For IPv6 ACLs, the switch supports one VACL assignment per VLAN. For IPv4 ACLs, the switch supports one inbound and one outbound RACL assignment per VLAN, and one VACL assignment per VLAN.

Syntax: show access-list vlan < *vid* >

Lists the current IPv4 and IPv6 ACL assignments to the specified VLAN (in the running config file).

Note

This information also appears in the **show running** output. If you execute **write memory** after configuring an ACL, it also appears in the **show config** output.

For example, the following output shows that all inbound IPv6 traffic and the inbound and outbound, routed IPv4 traffic are all filtered on VLAN 20.

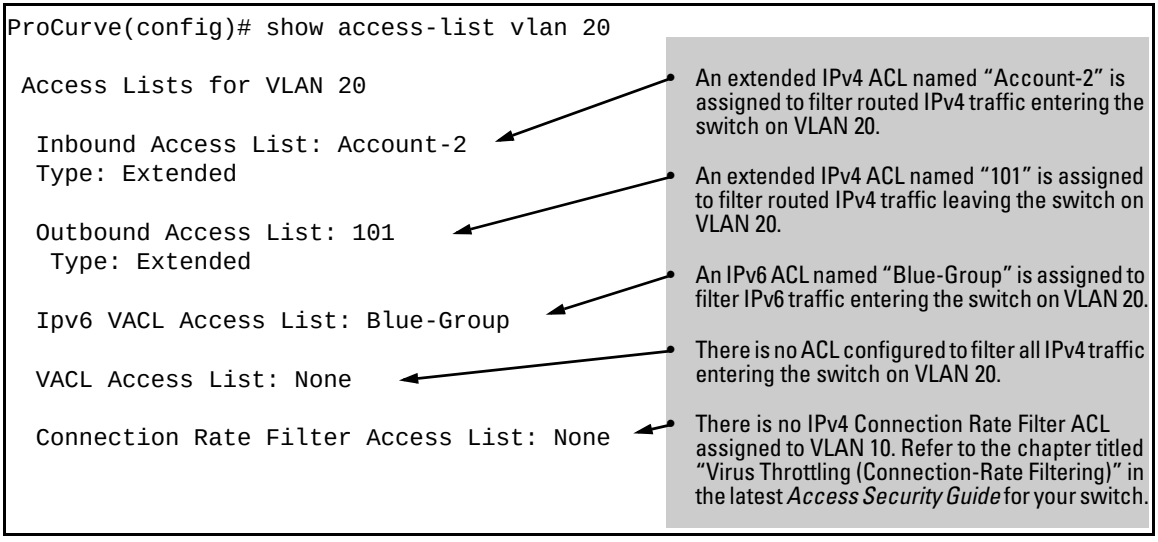
ProCurve(config)# show access-list vlan 20
Access Lists for VLAN 20
Inbound Access List: Account-2
Type: Extended
Outbound Access List: 101
Type: Extended
Ipv6 VACL Access List: Blue-Group
VACL Access List: None
Connection Rate Filter Access List: None
An extended IPv4 ACL named "Account-2" is assigned to filter routed IPv4 traffic entering the switch on VLAN 20.
An extended IPv4 ACL named "101" is assigned to filter routed IPv4 traffic leaving the switch on VLAN 20.
An IPv6 ACL named "Blue-Group" is assigned to filter IPv6 traffic entering the switch on VLAN 20.
There is no ACL configured to filter all IPv4 traffic entering the switch on VLAN 20.
There is no IPv4 Connection Rate Filter ACL assigned to VLAN 10. Refer to the chapter titled "Virus Throttling (Connection-Rate Filtering)" in the latest Access Security Guide for your switch.

Figure 8-28. Example of Listing the ACL Assignments for a VLAN

Display Static Port (and Trunk) ACL Assignments

This command lists the identification and type(s) of current static port ACL assignments to individual switch ports and trunks, as configured in the running-config file. (The switch allows one static port ACL assignment per port.)

Syntax: show access-list ports < all | *port-list* >

Lists the current static port ACL assignments for ports and trunks in the running config file.

Note

This information also appears in the **show running** output. If you execute **write memory** after configuring an ACL, it also appears in the **show config** output.

For example, the following output shows IPv4 and IPv6 ACLs configured on various ports and trunks on the switch:

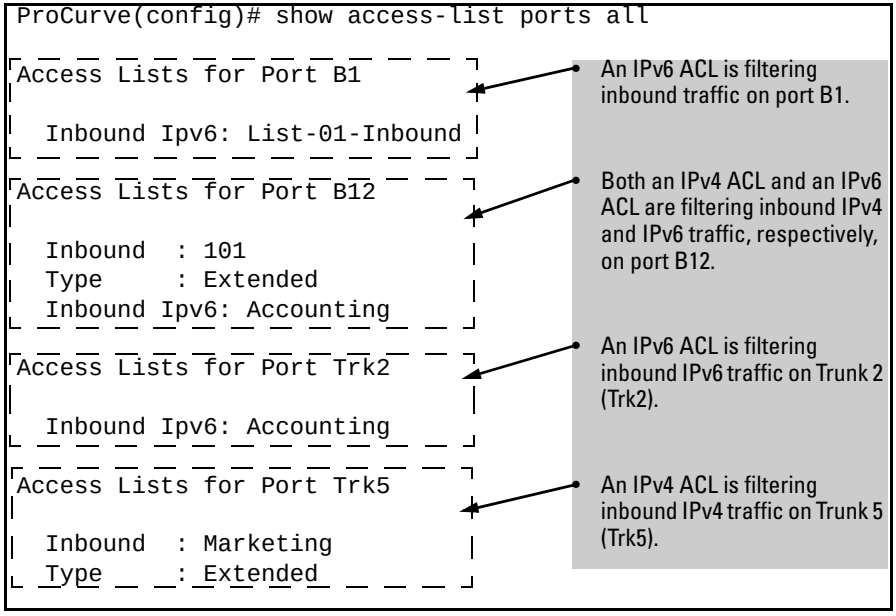
ProCurve(config)# show access-list ports all
Access Lists for Port B1
Inbound Ipv6: List-01-Inbound
Access Lists for Port B12
Inbound : 101
Type : Extended
Inbound Ipv6: Accounting
Access Lists for Port Trk2
Inbound Ipv6: Accounting
Access Lists for Port Trk5
Inbound : Marketing
Type : Extended
An IPv6 ACL is filtering inbound traffic on port B1.
Both an IPv4 ACL and an IPv6 ACL are filtering inbound IPv4 and IPv6 traffic, respectively, on port B12.
An IPv6 ACL is filtering inbound IPv6 traffic on Trunk 2 (Trk2).
An IPv4 ACL is filtering inbound IPv4 traffic on Trunk 5 (Trk5).

Figure 8-29. Example of Listing the ACL Assignments for Ports and Trunks

Displaying the Content of a Specific ACL

This command displays a specific IPv6 or IPv4 ACL configured in the running config file in an easy-to-read tabular format.

Note

This information also appears in the **show running** display. If you execute **write memory** after configuring an ACL, it also appears in the **show config** display.

For information on IPv4 ACL operation, refer to the latest version of the *Access Security Guide* for your switch.

Syntax: show access-list < *identifier* > [config]

Display detailed information on the content of a specific ACL configured in the running-config file.

For example, suppose you configured the following two ACLs in the switch:

Identifier	Type	Desired Action
Accounting	IPv6	• Permit Telnet traffic from these two IPv6 addresses: – 2001:db8:0:1af::10: 14 – 2001:db8:0:1af::10: 24 • Deny Telnet traffic from all other devices in the same subnet. • Permit all other IPv6 traffic from the subnet. • Deny and log any IPv6 traffic from any other source.
List-120	IPv4 Extended	• Permit any TCP traffic from 10.30.133.27 to any destination. • Deny any other IP traffic from 10.30.133.(1-255). • Permit all other IP traffic from any source to any destination.

Use **show access-list < *identifier* >** to inspect a specific IPv6 or IPv4 ACL, as follows:

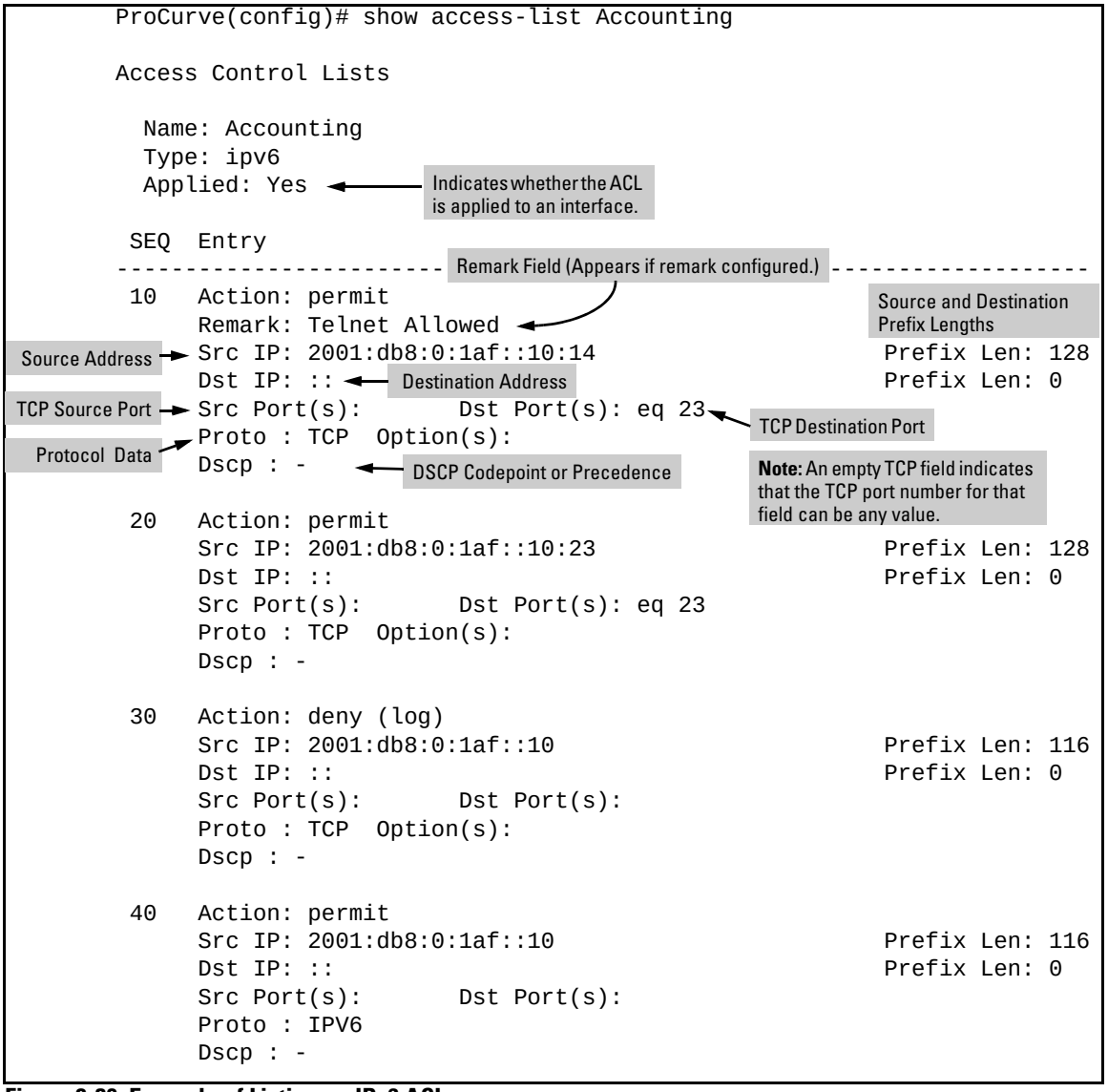

Figure 8-30. Example of Listing an IPv6 ACL

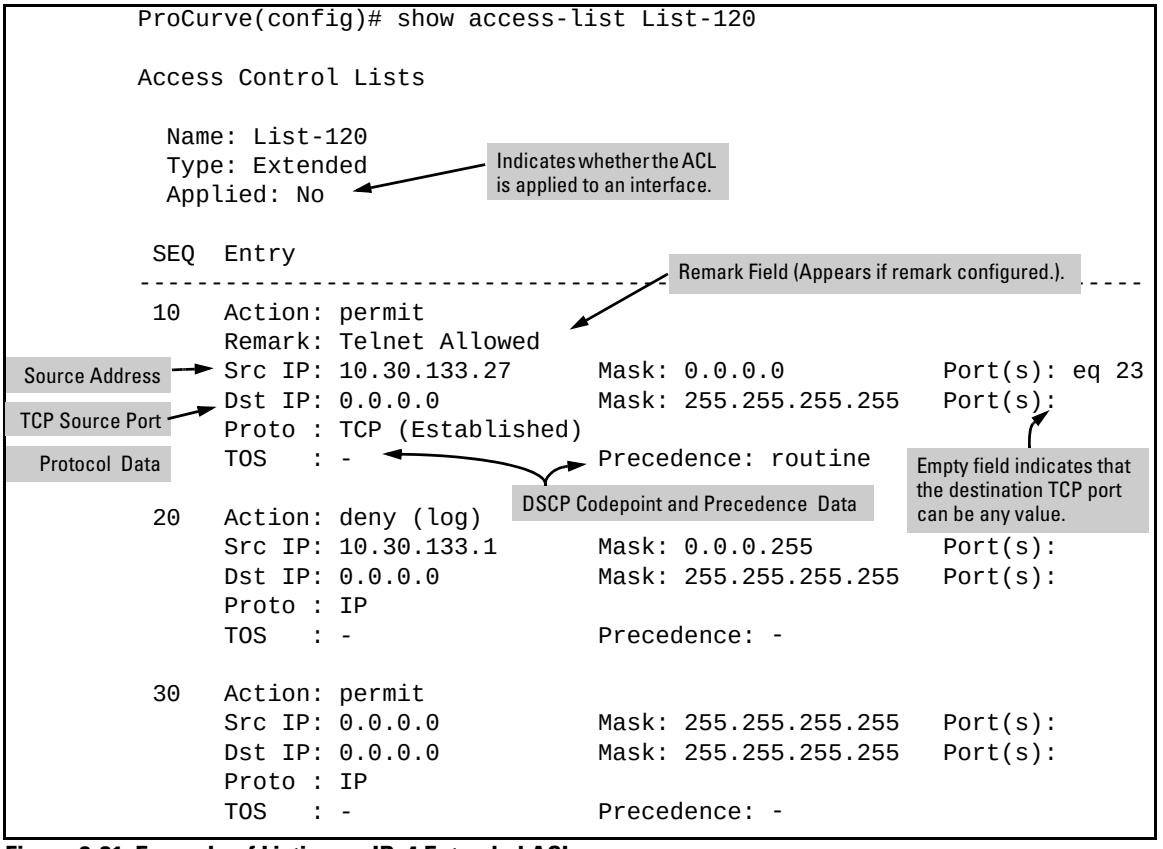
ProCurve(config)# show access-list List-120
Access Control Lists
Name: List-120
Type: Extended
Applied: No
Indicates whether the ACL is applied to an interface.
SEQ Entry
Remark Field (Appears if remark configured.).
10 Action: permit
Remark: Telnet Allowed
Source Address
Src IP: 10.30.133.27 Mask: 0.0.0.0 Port(s): eq 23
TCP Source Port
Dst IP: 0.0.0.0 Mask: 255.255.255.255 Port(s):
Proto : TCP (Established)
Protocol Data
TOS : - Precedence: routine
Empty field indicates that the destination TCP port can be any value.
DSCP Codepoint and Precedence Data
20 Action: deny (log)
Src IP: 10.30.133.1 Mask: 0.0.0.255 Port(s):
Dst IP: 0.0.0.0 Mask: 255.255.255.255 Port(s):
Proto : IP
TOS : - Precedence: -
30 Action: permit
Src IP: 0.0.0.0 Mask: 255.255.255.255 Port(s):
Dst IP: 0.0.0.0 Mask: 255.255.255.255 Port(s):
Proto : IP
TOS : - Precedence: -

Figure 8-31. Example of Listing an IPv4 Extended ACL

The **show access-list < *identifier* > config** command shows the same ACL data as **show access-list < *identifier* >** but in the format used by the **show < run | config >** commands to list the switch configuration. For example:

```
Port-1(config)# show access-list List-120 config

ip access-list extended "List-120"
   10 remark "Telnet Allowed"
   10 permit tcp 10.30.133.27 0.0.0.0 eq 23 0.0.0.0 255.255.255.255 precedence 0
 established
   20 deny ip 10.30.133.1 0.0.0.255 0.0.0.0 255.255.255.255 log
   30 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit
```

Figure 8-32. Example of an ACL Listed with the "Config" Option

Table 8-5. Descriptions of Data Types Included in Show Access-List < *acl-id* > Output

Field	Description
Name	The ACL identifier. For IPv6 ACLs, is an alphanumeric name. For IPv4 ACLs, can be a number from 1 to 199, or an alphanumeric name.
Type	IPv6, Standard, or Extended. IPv6 ACLs use a source and a destination address, plus IPv6 protocol specifiers. Standard ACLs are IPv4 only, and use only a source IP address. Extended ACLs are available in IPv4 only, and use both source and destination IP addressing, as well as other IP protocol specifiers.
Applied	"Yes" means the ACL has been applied to an interface. "No" means the ACL exists in the switch configuration, but has not been applied to any interface, and is therefore not in use.
SEQ	The sequential number of the Access Control Entry (ACE) in the specified ACL.
Entry	Lists the content of the ACEs in the selected ACL.
Action	Permit (forward) or deny (drop) a packet when it is compared to the criteria in the applicable ACE and found to match. Includes the optional **log** option, if used, in **deny** actions.
Remark	Displays any optional remark text configured for the selected ACE.
IP	**Used for IPv4 Standard ACEs:** The source IPv4 address to which the configured mask is applied to determine whether there is a match with a packet.
Src IP	**Used for IPv6 ACEs and IPv4 Extended ACEs:** The source IPv6 or IPv4 address to which the configured mask is applied to determine whether there is a match with a packet.
Dst IP	**Used for IPv6 ACEs and IPv4 Extended ACEs:** The source and destination IP addresses to which the corresponding configured masks are applied to determine whether there is a match with a packet.
Mask	Used in IPv4 ACEs, the mask is configured in an ACE and applied to the corresponding IP address in the ACE to determine whether a packet matches the filtering criteria.
Prefix Len (source and destination)	Used in IPv6 ACEs to specify the number of consecutive high-order (leftmost) bits of the source and destination addresses configured in an ACE to be used to determine a match with a packet being filtered by the ACE.
Proto	Used in IPv6 ACEs and IPv4 extended ACEs to specify the packet protocol type to filter.
Port(s)	Used in IPv4 extended ACEs to show any TCP or UDP operator and port number(s) included in the ACE.
Src Port(s) Dst Port(s)	Used in IPv6 ACEs to show TCP or UDP source and destination operator and port number(s) included in the ACE.
DSCP	Used in IPv6 ACEs to show the DSCP precedence or codepoint setting, if any.
TOS	Used in IPv4 extended ACEs to indicate Type-of-Service setting, if any.
Precedence	Used in IPv4 extended ACEs to indicate the IP precedence setting, if any.

Display All ACLs and Their Assignments in the Switch Startup-Config File and Running-Config File

The **show config** and **show running** commands include in their listings any configured ACLs and any ACL assignments to VLANs. Refer to figure 8-36 (page 8-91) for an example. Remember that **show config** lists the startup-config file and **show running** lists the running-config file.

Creating or Editing ACLs Offline

The section titled “Editing an Existing ACL” on page 8-66 describes how to use the CLI to edit an ACL, and is most applicable in cases where the ACL is short or there is only a minor editing task to perform. The offline method provides a useful alternative to using the CLI for creating or extensively editing a large ACL.This section describes how to:

- move an existing ACL to a TFTP server
- use a text (.txt) file format to create a new ACL or edit an existing ACL offline
- use TFTP to load an offline ACL into the switch’s running-config

For longer ACLs that may be difficult or time-consuming to accurately create or edit in the CLI, you can use the offline method described in this section.

Note

Beginning with software release K_12_XX, **copy** commands that used either **tftp** or **xmodem**, also include an option to use **usb** as a source or destination device for file transfers. So although the following example highlights tftp, remember that **xmodem** or **usb** can also be used to transfer ACLs to and from the switch.

Creating or Editing an ACL Offline

The Offline Process

1. Begin by doing one of the following:
   - To edit one or more existing ACLs, use **copy command-output tftp** to copy the current version of the ACL configuration to a file in your TFTP server. For example, to copy the ACL configuration to a file named **acl-001.txt** in the TFTP directory on a server at FE80::2a1:200.

     ```
     ProCurve# copy command-output 'show access-list
     config' tftp fe80::2a1:200 acl-001.txt pc
     ```
   - To create a new ACL, open a text (.txt) file in the appropriate directory on a TFTP server accessible to the switch.
2. Use a text editor to create or edit the ACL(s) in the ***.txt** ASCII file format.

If you are replacing an ACL on the switch with a new ACL that uses the same number or name syntax, begin the command file with a **no ip access-list** command to remove the earlier version of the ACL from the switch's running-config file. Otherwise, the switch will append the new ACEs in the ACL you download to the existing ACL. For example, if you planned to use the **copy** command to *replace* an ACL named "List-120", you would place this command at the beginning of the edited file:

```
no ipv6 access-list List-120
```

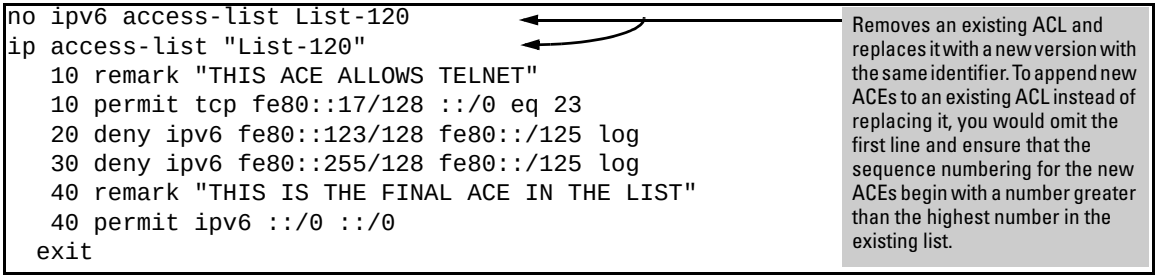
no ipv6 access-list List-120
ip access-list "List-120"
10 remark "THIS ACE ALLOWS TELNET"
10 permit tcp fe80::17/128 ::/0 eq 23
20 deny ipv6 fe80::123/128 fe80::/125 log
30 deny ipv6 fe80::255/128 fe80::/125 log
40 remark "THIS IS THE FINAL ACE IN THE LIST"
40 permit ipv6 ::/0 ::/0
exit
Removes an existing ACL and replaces it with a new version with the same identifier. To append new ACEs to an existing ACL instead of replacing it, you would omit the first line and ensure that the sequence numbering for the new ACEs begin with a number greater than the highest number in the existing list.

Figure 8-33. Example of an Offline ACL File Designed To Replace An Existing ACL

3. Use **copy tftp command-file** to download the file as a list of commands to the switch.

Example of Using the Offline Process

For example, suppose that you wanted to create an IPv6 ACL for a VACL application and download it to a switch from a TFTP server at FE80::1ad:17.

1. You would create a **.txt** file with the content shown in figure 8-34.

```
ipv6 access-list "acl-001"

; CREATED ON JUNE 10

10 remark "Telnet Allowed Here"
10 permit tcp 2001:db8:0:1af::10:14/128 ::/0 eq 23
20 permit tcp 2001:db8:0:1af::10:23/128 ::/0 eq 23
30 deny tcp 2001:db8:0:1af::10/116 ::/0 log
40 permit ipv6 2001:db8:0:1af::10/116 ::/0
45 permit ipv6 2001:db8:0:2b1::/64 ::/0
50 deny ipv6 ::/0 ::/0 log
exit
vlan 20 ipv6 access-group acl-001 vlan
```

The "; " enables a comment in the file.

Note: You can use the "; " character to denote a comment. The file stored on your TFTP server retains comments, and they appear when you use **copy** to download the ACL command file. (Comments are not saved in the switch configuration.)

Figure 8-34. Example of a .txt File Designed for Creating an ACL

2. After you copy the above .txt file to the TFTP server at FE80::1ad:17, you would then execute the following command:

   copy tftp command-file fe80::1ad:17 acl-001.txt pc

In this example, the CLI would show output similar to the following to indicate that the ACL was successfully downloaded to the switch:

Note If a transport error occurs, the switch does not execute the command and the ACL is not configured.

```
ProCurve(config)# copy tftp command-file fe80::1ad:17 acl-001.txt pc
Running configuration may change, do you want to continue [y/n]?  y
  1. ipv6 access-list "acl-001"
  6.    ; CREATED ON JUNE 10
 10.    10 remark "Telnet Denied Here"
 13.    10 deny tcp 2001:db8:0:1af::/64 ::/0 eq 23
 16.    30 deny tcp ::/0 ::/0 log
 19.    40 deny icmp 2001:db8:0:1af::/64 ::/0 134
 22.    50 deny icmp 2001:db8:0:1af::/64 ::/0 133
 27.    ; PERMITS IPV6 ANY ANY
 31.    60 permit ipv6 ::/0 ::/0
 34.    exit
 36.    vlan 20 ipv6 access-group acl-001 vlan
```

Note: Blank lines may appear in the command output when you copy the command file to the switch. However, they are eliminated in the copy of the ACL in switch memory. This is normal operation. (See also figure 8-36 for the configuration resulting from this output.)

Figure 8-35. Example of Using "copy tftp command-file" To Configure an ACL in the Switch

3. In this example, the command to assign the ACL to a VLAN was included in the .txt command file. If this is not done in your applications, then the next step is to manually assign the new ACL to the intended VLAN.

   vlan < *vid* > ipv6 access-group < *identifier* > vlan

4. You can then use the **show run** or **show access-list config** command to inspect the switch configuration to ensure that the ACL was properly downloaded.

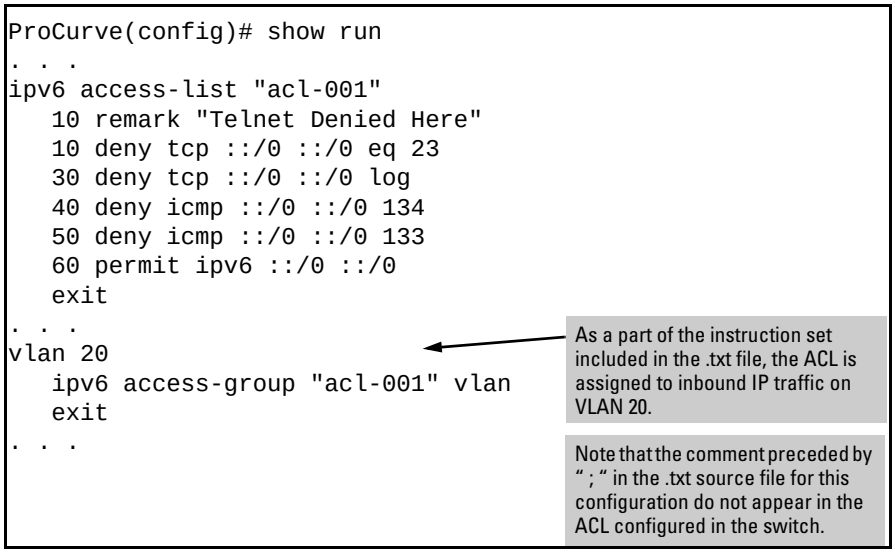
ProCurve(config)# show run
. . .
ipv6 access-list "acl-001"
10 remark "Telnet Denied Here"
10 deny tcp ::/0 ::/0 eq 23
30 deny tcp ::/0 ::/0 log
40 deny icmp ::/0 ::/0 134
50 deny icmp ::/0 ::/0 133
60 permit ipv6 ::/0 ::/0
exit
. . .
vlan 20
ipv6 access-group "acl-001" vlan
exit
. . .
As a part of the instruction set included in the .txt file, the ACL is assigned to inbound IP traffic on VLAN 20.
Note that the comment preceded by " ; " in the .txt source file for this configuration do not appear in the ACL configured in the switch.

Figure 8-36. Example of Verifying the .txt File Download to the Switch

5. If the configuration appears satisfactory, save it to the startup-config file:

```
ProCurve(config)# write memory
```

Testing and Troubleshooting ACLs

You can monitor ACL performance by using the "Deny" logging option (which generates log messages when there is a "deny" ACE match) and the ACE statistics counters (which maintain running totals of the packet matches on each ACE in an ACL).

Enable IPv6 ACL "Deny" Logging

ACL logging enables the switch to generate a message when IP traffic meets the criteria for a match with an ACE that results in an explicit "deny" action. You can use ACL logging to help:

- Test your network to help ensure that your ACL configuration is detecting and denying the incoming IPv6 traffic you do not want to enter the switch.
- Receive notification when the switch denies inbound IPv6 traffic you have designed your ACLs to reject (deny).

The switch sends ACL messages to Syslog and optionally to the current console, Telnet, or SSH session. You can use **logging < >** to configure up to six Syslog server destinations.

Requirements for Using IPv6 ACL Logging

- The switch configuration must include an ACL (1) assigned to a port, trunk, or static VLAN interface and (2) containing an ACE configured with the **deny** action and the **log** option.
- For IPv6 ACL logging to a Syslog server:
  - The server must be accessible to the switch and identified in the running configuration.
  - The logging facility must be enabled for Syslog.
  - Debug must be configured to:
    - support ACL messages
    - send debug messages to the desired debug destination

  These requirements are described in more detail under "Enabling ACL Logging on the Switch" on page 8-93.

ACL Logging Operation

When the switch detects a packet match with an ACE and the ACE includes both the **deny** action and the optional **log** parameter, an ACL log message is sent to the designated debug destination. The first time a packet matches an ACE with **deny** and **log** configured, the message is sent immediately to the destination and the switch starts a wait-period of approximately five minutes. (The exact duration of the period depends on how the packets are internally routed.) At the end of the collection period, the switch sends a single-line summary of any additional "deny" matches for that ACE (and any other "deny" ACEs for which the switch detected a match). If no further log messages are generated in the wait-period, the switch suspends the timer and resets itself to send a message as soon as a new "deny" match occurs. The data in the message includes the information illustrated in figure 8-37.

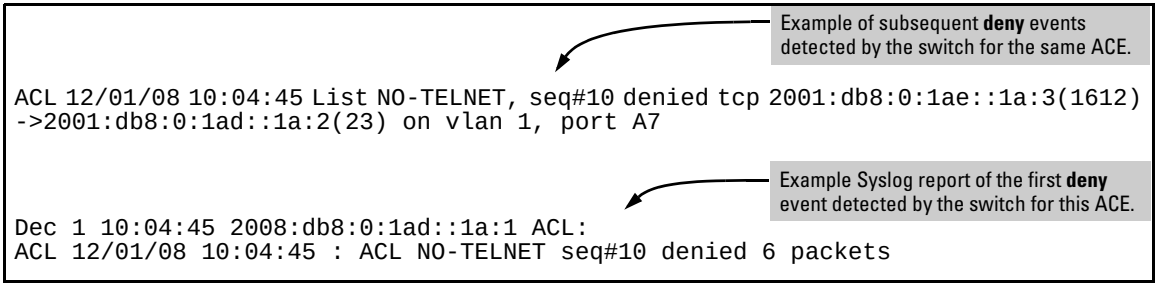
Example of subsequent **deny** events detected by the switch for the same ACE.
ACL 12/01/08 10:04:45 List NO-TELNET, seq#10 denied tcp 2001:db8:0:1ae::1a:3(1612)
->2001:db8:0:1ad::1a:2(23) on vlan 1, port A7
Example Syslog report of the first **deny** event detected by the switch for this ACE.
Dec 1 10:04:45 2008:db8:0:1ad::1a:1 ACL:
ACL 12/01/08 10:04:45 : ACL NO-TELNET seq#10 denied 6 packets

Figure 8-37. Content of a Message Generated by an ACL-Deny Action

Enabling ACL Logging on the Switch

1. If you are using a Syslog server, use the **logging < *ip-addr* >** command to configure the Syslog server IP address(es). Ensure that the switch can access any Syslog server(s) you specify.
2. Use **logging facility syslog** to enable the logging for Syslog operation.
3. Use the **debug destination** command to configure one or more log destinations. (Destination options include **logging** and **session**. For more information on debug, refer to "Debug and Syslog Messaging Operation" in appendix C, "Troubleshooting", in the latest *Management and Configuration Guide* for your switch.)
4. Use **debug acl** or **debug all** to configure the debug operation to include ACL messages.
5. Configure an ACL with the **deny** action and the **log** option in one or more ACEs.

For example, suppose that you want to configure the following on a switch receiving IPv6 traffic and configured for IPv4 routing:

- For port B1 on VLAN 10 configure an IPv6 ACL with an ACL-ID of "NO-TELNET" and use the PACL **in** option to deny Telnet traffic entering the switch from IP address FE80::10:3.
- Configure the switch to send an ACL log message to the current console session and to a Syslog server at 10.10.50.173 on VLAN 50 if the switch detects a packet match denying a Telnet attempt from FE80::10:3.

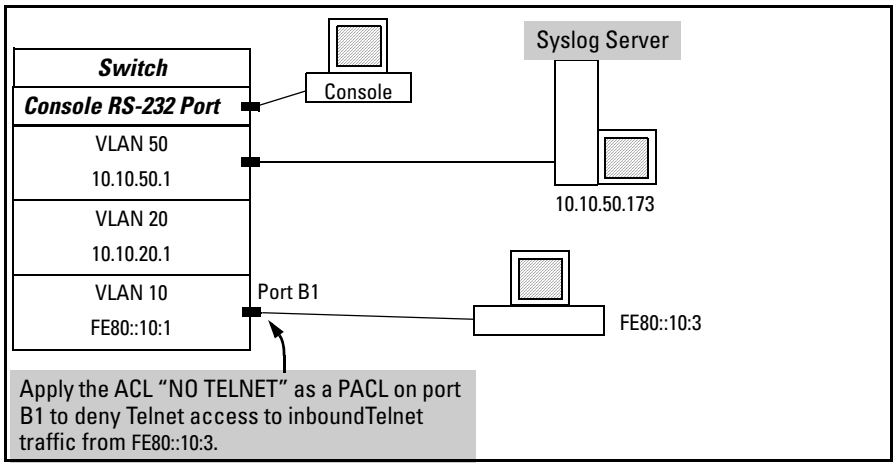

Figure 8-38. Example of an ACL Log Application

```
ProCurve(config)# ipv6 access-list NO-TELNET
ProCurve(config-ipv6-acl)# remark "deny fe80::10:3 Telnet traffic."
ProCurve(config-ipv6-acl)# deny tcp host fe80::10:3 any eq telnet log
ProCurve(config-ipv6-acl)# permit ipv6 any any
ProCurve(config-ipv6-acl)# exit
ProCurve(config)# vlan 10 ipv6 access-group NO-TELNET vlan
ProCurve(config)# logging 10.10.50.173
ProCurve(config)# logging facility syslog
ProCurve(config)# debug destination logging
ProCurve(config)# debug destination session
ProCurve(config)# debug acl
ProCurve(config)# write mem
ProCurve(config)# show debug

 Debug Logging

  Destination:
   Logging --
     10.10.50.173
     Facility = syslog
     Severity = debug
     System Module = all-pass
     Priority Desc =
   Session

  Enabled debug types:
   event
   acl log
ProCurve(config)# show access-list NO-TELNET config

ipv6 access-list "NO-TELNET"
   10 remark "deny fe80::10:3 TELNET TRAFFIC"
   10 deny tcp fe80::10:3/128 ::/0 eq 23 log
   20 permit ipv6 ::/0 ::/0
 exit
```

Assigns the ACL named "NO-TELNET" as a VACL to filter Telnet traffic from FE80::10:3 entering the switch on VLAN 10.

Figure 8-39. Commands for Applying an ACL with Logging to Figure 8-38

Monitoring Static ACL Performance

ACL statistics counters provide a means for monitoring ACL performance by using counters to display the current number of matches the switch has detected for each ACE in an ACL assigned to a switch interface. This can help, for example, to determine whether a particular traffic type is being filtered by the intended ACE in an assigned list, or if traffic from a particular device or network is being filtered as intended.

Note

This section describes the command for monitoring static ACL performance. To monitor RADIUS-assigned ACL performance, use either of the following commands:

show access-list radius < all | *port-list* >

show port-access < authenticator | mac-based | web-based > clients < *port-list* > detailed

Refer to the chapter titled “Configuring RADIUS Server Support for Switch Services” in the latest Access Security Guide for your switch.

Syntax: show statistics
aclv4 < *acl-name-str* > port < *port-#* >
aclv4 < *acl-name-str* > vlan < *vid* > < in | out | vlan >

aclv6 < *acl-name-str* > port < port-# >
aclv6 < *acl-name-str* > vlan < *vid* > vlan

Displays the current match (hit) count per ACE for the specified IPv6 or IPv4 static ACL assignment on a specific interface:

Total: *This column lists the running total of the matches the switch has detected for the ACEs in an applied ACL since the ACL's counters were last reset to 0 (zero)*

For example figure 8-40 illustrates both IPv6 and IPv4 ACL activity:

```
ProCurve# show statistics aclv6 IPV6-ACL vlan 20 vlan

 HitCounts for ACL IPV6-ACL

  Total

(     12)    10 permit icmp ::/0 fe80::20:2/128 128
(      6)    20 deny tcp ::/0 fe80::20:2/128 eq 23 log
(     41)    30 permit ipv6 ::/0 ::/0

ProCurve# show statistics aclv4 102 vlan 20 vlan

 HitCounts for ACL 102

  Total

(      4)    10 permit icmp 10.10.20.3 0.0.0.0 10.10.20.2 0.0.0.0 8
(      8)    20 deny icmp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 8
(      2)    30 permit tcp 10.10.20.3 0.0.0.255 10.10.20.2 0.0.0.255 eq 23
(      2)    55 deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 8
(    125)    60 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Figure 8-40. Example of IPv6 and IPv4 ACL Statistics

ACE Counter Operation: *For a given ACE in an assigned ACL, the counter increments by 1 each time the switch detects a packet that matches the criteria in that ACE, and maintains a running total of the matches since the last counter reset.*

For example, in ACL line 10 below, there has been a total of 37 matches on the ACE since the last time the ACL's counters were reset.

```
 Total

(    37)    10 permit icmp ::/0 fe80::20:2/128 128
```

Notes: *This ACL monitoring feature does not include hits on the "implicit deny" that is included at the end of all ACLs.*

Also, if the **show statistics** *command does not show any ACE hit activity at first use, re-execute the command.*

Resetting ACE Hit Counters to Zero:

- *Removing an ACL from an interface zeros the ACL's ACE counters for that interface only.*
- *For a given ACL, either of the following actions clear the ACE counters to zero for all interfaces to which the ACL is assigned.*
  - *adding or removing a permit or deny ACE in the ACL*
  - *rebooting the switch*

Example of ACL Performance Monitoring

Figure 8-41 shows a sample of performance monitoring output for an IPv6 ACL assigned as a VACL.

```
ProCurve# show statistics aclv6 V6-02 vlan 20 vlan

 HitCounts for ACL V6-02

  Total

(      5)    10 permit icmp ::/0 fe80::20:2/128 128
(      4)    20 permit icmp ::/0 fe80::20:3/128 128
(    136)    30 permit tcp fe80::20:1/128 ::/0 eq 23
(      2)    40 deny icmp ::/0 fe80::20:1/128 128
(     10)    50 deny tcp ::/0 ::/0 eq 23
(      8)    60 deny icmp ::/0 ::/0 133
(    155)    70 permit ipv6 ::/0 ::/0
```

Figure 8-41. Example of IPv6 ACL Performance Monitoring Output

Figure 8-42 shows a sample of performance monitoring output for an IPv4 ACL assigned as a VACL.

```
ProCurve# show statistics aclv4 102 vlan 20 vlan

 HitCounts for ACL 102

  Total

(      1)    10 permit icmp 10.10.20.3 0.0.0.0 10.10.20.2 0.0.0.0 8
(      2)    20 deny icmp 10.10.20.3 0.0.0.0 10.10.20.1 0.0.0.0 8 log

(      2)    30 deny icmp 10.10.20.2 0.0.0.0 10.10.20.3 0.0.0.0 8 log

(      1)    40 deny icmp 10.10.20.2 0.0.0.0 10.10.20.1 0.0.0.0 8 log

(     10)    50 deny tcp 10.10.20.2 0.0.0.255 10.10.20.3 0.0.0.255 eq 23 log
(     27)    60 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Figure 8-42. Example of IPv4 ACL Performance Monitoring Output

IPv6 Counter Operation with Multiple Interface Assignments

Note

The examples of counters in this section use small values to help illustrate counter operation. The counters in real-time network applications are generally much more active and show higher values.

Where the same IPv6 ACL is assigned to multiple interfaces, the switch maintains a separate instance of each ACE counter in the ACL. When there is a match with traffic on one of the ACL's assigned interfaces, only the affected ACE counters for that interface are incremented. Other instances of the same ACL applied to other interfaces are not affected.

For example, suppose that:

- An ACL named "V6-01" is configured as shown in figure 8-43 to block Telnet access to a workstation at FE80::20:2, which is connected to a port belonging to VLAN 20.
- The ACL is assigned as a PACL (port ACL) on port B2, which is also a member of VLAN 20:

```
ProCurve(config)# show access-list V6-01 config

ipv6 access-list "V6-01"
     10 permit icmp ::/0 fe80::20:2/128 128
     20 deny tcp ::/0 fe80::20:2/128 eq 23 log
     30 permit ipv6 ::/0 ::/0
   exit

ProCurve(config)# int b2 ipv access-group V6-01 in
```

Assigns the ACL to port B2.

Figure 8-43. ACL "V6-01" and Command for PACL Assignment on Port B2

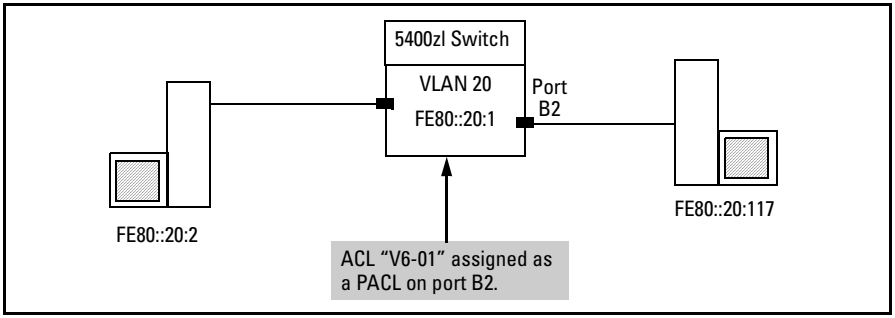

Figure 8-44. Application to Filter Traffic Inbound on Port B2

Using the topology in figure 8-44, a workstation at FE80::20:117 on port B2 attempting to ping and Telnet to the workstation at FE80::20:2 is filtered through the PACL instance of the "V6-01" ACL assigned to port B2, resulting in the following:

```
ProCurve# ping6 fe80::20:2%vlan20
fe80:0000:0000:0000:0000:0000:0020:0002 is alive, time = 5 ms
ProCurve# telnet fe80::20:2%vlan20
Telnet failed: Connection timed out.
ProCurve#
```

Figure 8-45. Ping and Telnet from FE80::20:117 to FE80::20:2 Filtered by the Assignment of "V6-01" as a PACL on Port B2

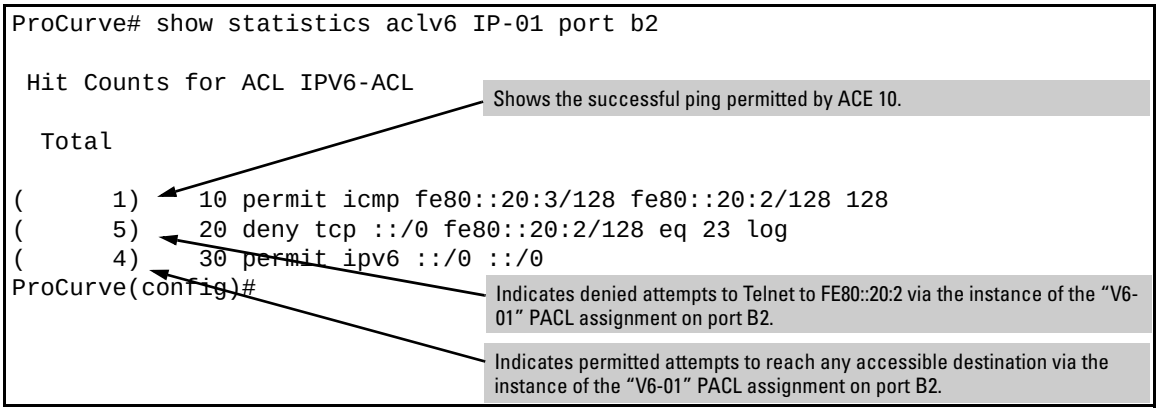
ProCurve# show statistics aclv6 IP-01 port b2
Hit Counts for ACL IPV6-ACL
Total
(1) 10 permit icmp fe80::20:3/128 fe80::20:2/128 128
(5) 20 deny tcp ::/0 fe80::20:2/128 eq 23 log
(4) 30 permit ipv6 ::/0 ::/0
ProCurve(config)#
Shows the successful ping permitted by ACE 10.
Indicates denied attempts to Telnet to FE80::20:2 via the instance of the "V6-01" PACL assignment on port B2.
Indicates permitted attempts to reach any accessible destination via the instance of the "V6-01" PACL assignment on port B2.

Figure 8-46. Resulting ACE Hits on ACL "V6-01"

Note IPv4 ACE counters assigned as RACLs operate differently than described above. For more information, refer to the following section.

IPv4 Counter Operation with Multiple Interface Assignments

Where the same IPv4 ACL is assigned to multiple interfaces as a VLAN ACL (VACL) or port ACL (PACL), the switch maintains a separate instance of ACE counters for each interface assignment. Thus, when there is a match with traffic on one of the ACL's VACL- or PACL -assigned interfaces, only the ACE counter in the affected instance of the ACL is incremented. However, if an ACL has multiple assignments as an RACL, then a match with an ACE in any RACL instance of the ACL increments that same counter on all RACL-assigned instances of that ACL. (The ACE counters for VACL and PACL instances of an ACL are not affected by counter activity in RACL instances of the same ACL.)

For example, suppose that an IPv4 ACL named "Test-1" is configured as shown in figure 8-47 to block Telnet access to a server at 10.10.20.12 on VLAN 20, and that the Test-1 ACL is assigned to VLANs as follows:

- VLAN 20: VACL
- VLAN 50: RACL
- VLAN 70: RACL

```
ProCurve(config)# show access-list Test1 config

ip access-list extended “Test1”
   10 deny tcp 0.0.0.0 255.255.255.255 10.10.20.12 0.0.0.0 eq 23 log
   20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
   exit

ProCurve(config)# vlan 20 ip access-group Test-1 vlan
ProCurve(config)# vlan 50 ip access-group Test-1 in
ProCurve(config)# vlan 70 ip access-group Test-1 in
```

Assigns the ACL as a VACL to VLAN 20.

Assigns the ACL as an RACL to VLANs 50 and 70.

Figure 8-47. ACL “Test-1” and Interface Assignment Commands

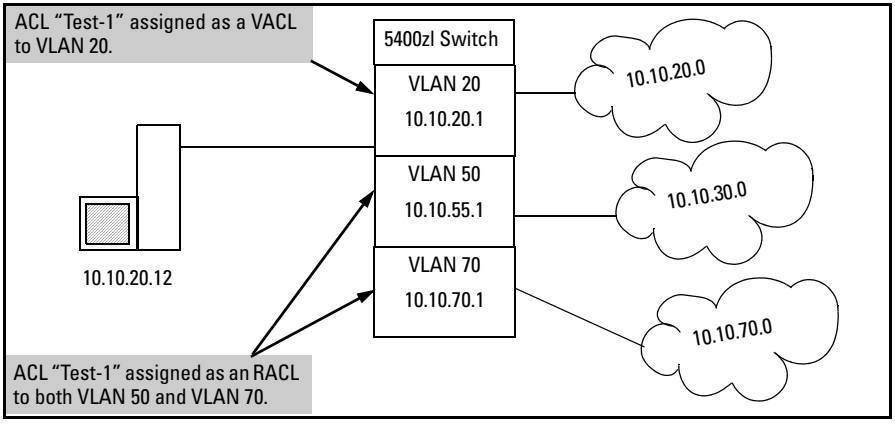

Figure 8-48. Example of Using the Same IPv4 ACL for VACL and RACL Applications

In the above case:

- Matches with ACEs 10 or 20 that originate on VLAN 20 will increment only the counters for the instances of these two ACEs in the Test-1 VACL assignment on VLAN 20. The same counters in the instances of ACL Test-1 assigned to VLANs 50 and 70 will not be incremented.
- Any Telnet requests to 10.10.20.12 that originate on VLANs 50 or 70 will be filtered by instances of Test-1 assigned as RACLs, and will increment the counters for ACE 10 on both RACL instances of the Test-1 ACL.

Using the network in figure 8-48, a device at 10.10.20.4 on VLAN 20 attempting to ping and Telnet to 10.10.20.12 is filtered through the VACL instance of the “Test-1” ACL on VLAN 20 and results in the following:

```
ProCurve(config)# ping 10.10.20.2
10.10.20.2 is alive, time = 5 ms
ProCurve(config)# telnet 10.10.20.2
Telnet failed: Connection timed out.
ProCurve(config)#
```

Figure 8-49. Ping and Telnet from 10.10.20.4 to 10.10.20.2 Filtered by the Assignment of "Test-1" as an IPv4 VACL on VLAN 20

```
ProCurve(config)# show statistics aclv4 Test-1 vlan 20 vlan

 Hit Counts for ACL Test-1

  Total

(     5)    10 deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log
(     2)    20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255

ProCurve# show statistics aclv4 Test-1 vlan 50 in

 Hit Counts for ACL Test-1

  Total

(      0)    10 deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log
(      0)    20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
```

Indicates denied attempts to Telnet to 10.10.20.12 filtered by the instance of the "Test-1" VACL assignment on VLAN 20.

Indicates permitted attempts to reach any accessible destination via the instance of the "Test-1" VACL assignment on VLAN 20. In this example, shows the successful pings permitted by ACE

Shows that the hits on the instance of the "Test-1" VACL assignment on VLAN 20 have no effect on the counters for the RACL assignment of "Test-1" on VLAN 50.

Figure 8-50. Resulting ACE Hits on IPv4 ACL "Test-1"

However, using a device at 10.10.30.11 on VLAN 50 for attempts to ping and Telnet to 10.10.20.12 requires routing, and filters the attempts through the RACL instance of the "Test-1" ACL on VLAN 50.

```
ProCurve# ping 10.10.20.2
10.10.20.2 is alive, time = 25 ms
ProCurve# telnet 10.10.20.2
Telnet failed: Connection timed out.
ProCurve#
```

Figure 8-51. Ping and Telnet from 10.10.30.11 to 10.10.20.2 Filtered by the Assignment of "Test-1" as an IPv4 RACL on VLAN 30

This action has an identical effect on the counters in all RACL instances of the "Test-1" ACL configured and assigned to interfaces on the same switch. In this example, it means that the RACL assignments of "Test-1" on VLANs 50 and 70 will be incremented by the above action occurring on VLAN 50.

```
ProCurve(config)# show statistics aclv4 Test-1 vlan 50 in

 Hit Counts for ACL Test-1

  Total

(      6)    10 deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log
(      1)    20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
ProCurve(config)#
```

Indicates the same type of data as shown in figure 8-50 for the VACL assignment of the "Test-1" ACL. That is, the Ping attempt incremented the counters for ACE 20 and the Telnet attempt incremented the counters for ACE 10 in the VLAN 50 RACL instance of the ACL.

Figure 8-52. Resulting ACE Hits on the VLAN 30 IPv4 RACL Assignment of the "Test-1" ACL

```
ProCurve(config)# show statistics aclv4 Test-1 vlan 70 in

 HitCounts for ACL Test-1

  Total

(      6)    10 deny tcp 0.0.0.0 255.255.255.255 10.10.20.2 0.0.0.0 eq 23 log
(      1)    20 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
ProCurve(config)#
```

The ACE counters in the VLAN 70 RACL assignment of "Test-1" are also incremented by the commands executed in figure 8-51.

Figure 8-53. Resulting ACE Hits on the VLAN 70 IPv4 RACL Assignment of the "Test-1" ACL

Note that the ACE counters for the *VACL* assignment of the "Test-1" ACL on VLAN 20 are not affected by ACE hits on the RACL assignments of the same ACL.

General ACL Operating Notes

ACLs do not provide DNS hostname support. ACLs cannot be configured to screen hostname IP traffic between the switch and a DNS.

ACLs Do Not Affect Serial Port Access. ACLs do not apply to the switch's serial port.

ACL Logging.

- The ACL logging feature generates a message only when packets are explicitly denied as the result of a match, and not when explicitly permitted or implicitly denied. To help test ACL logging, configure the last entry in an ACL as an explicit **deny** statement with a **log** statement included, and apply the ACL to an appropriate port or VLAN.
- Logging enables you to selectively test specific devices or groups. However, excessive logging can affect switch performance. For this reason, ProCurve recommends that you remove the logging option from ACEs for which you do not have a present need. Also, avoid configuring logging where it does not serve an immediate purpose. (Note that ACL logging is not designed to function as an accounting method.) See also "Apparent Failure To Log All 'Deny' Matches" in the section titled "ACL Problems", found in appendix C, "Troubleshooting" of the latest *Management and Configuration Guide* for your switch.
- When configuring logging, you can reduce excessive resource use by configuring the appropriate ACEs to match with specific hosts instead of entire subnets. (For more on resource usage, refer to "Monitoring Shared Resources" on page 8-105.)

Minimum Number of ACEs in an IPv6 ACL. An IPv6 ACL must include at least one ACE to enable traffic screening. An IPv6 ACL can be created "empty"; that is, without any ACEs. However if an empty ACL applied to an interface, the Implicit Deny function does not operate, and the ACL has no effect on traffic.

Monitoring Shared Resources. Applied ACLs share internal switch resources with several other features. However, if the internal resources become fully subscribed, additional ACLs cannot be applied until the necessary resources are released from other applications. For information on determining current resource availability and usage, refer to appendix E,

"Monitoring Resources" in the latest *Management and Configuration Guide* for your switch. See also the appendix titled "Scalability and System Maximums" in the same guide.

Protocol Support. ACL criteria does not include use of MAC address information or QoS.

Replacing or Adding To an Active IPv6 ACL Policy. If you assign an IPv6 ACL to an interface and subsequently add or replace ACEs in that ACL, each new ACE becomes active when you enter it. If the ACL is configured on multiple interfaces when the change occurs, then the switch resources must accommodate all applications of the ACL. If there are insufficient resources to accommodate one of several ACL applications affected by the change, then the change is not applied to any of the interfaces and the previous version of the ACL remains in effect. Refer to "Monitoring Shared Resources", above.

"Strict" IPv6 TCP and UDP. When the IPv6 ACL configuration includes TCP or UDP options, the switch operates in "strict" TCP and UDP mode for increased control. In this case, the switch compares all IPv6 TCP and UDP packets against the IPv6 ACLs.

Connection-Rate ACLs. As of software release K.14.01, this ACL connection-rate ACLs are supported for IPv4 ACLs, but not for IPv6 ACLs.

Unable to Delete an Empty ACL in the Running Configuration. The **no vlan < *vid* > ipv6 access-group < *name-str* > vlan** command does not delete the named ACL if the ACL is currently assigned to an interface.

9

IPv6 Diagnostic and Troubleshooting

Contents

Introduction 9-2

ICMP Rate-Limiting 9-2

Ping for IPv6 (Ping6) 9-4

Traceroute for IPv6 9-6

DNS Resolver for IPv6 9-9

- DNS Configuration 9-9
- Viewing the Current Configuration 9-11
- Operating Notes 9-11

Debug/Syslog for IPv6 9-12

- Configuring Debug and Event Log Messaging 9-12
- Debug Command 9-13
- Configuring Debug Destinations 9-15
- Logging Command 9-16

Introduction

Feature	Default	CLI
IPv6 ICMP Message Interval and Token Bucket	100 ms 10 max tokens	9-3
ping6	Enabled	
traceroute6	n/a	

The IPv6 ICMP feature enables control over the error and informational message rate for IPv6 traffic, which can help mitigate the effects of a Denial-of-service attack. Ping6 enables verification of access to a specific IPv6 device, and traceroute6 enables tracing the route to an IPv6-enabled device on the network.

ICMP Rate-Limiting

ICMP rate-limiting controls the rate at which ICMPv6 generates error and informational messages for features such as:

- neighbor solicitations
- neighbor advertisements
- multicast listener discovery (MLD)
- path MTU discovery (PMTU)
- duplicate address discovery (DAD)
- neighbor unreachability detection (NUD)
- router discovery
- neighbor discovery (NDP)

ICMPv6 error message generation is enabled by default. The rate of message generation can be adjusted, or message generation can be disabled.

Controlling the frequency of ICMPv6 error messages can help to prevent DoS (Denial- of- Service) attacks. With IPv6 enabled on the switch, you can control the allowable frequency of these messages with ICMPv6 rate-limiting.

Syntax:. ipv6 icmp error-interval < 0 - 2147483647 > [bucket-size < 1 - 200 >]
no ipv6 icmp error-interval

This command is executed from the global configuration level, and uses a "token bucket" method for limiting the rate of ICMP error and informational messages. Using this method, each ICMP message uses one token, and a message can be sent only if there is a token available. In the default configuration, a new token can be added every 100 milliseconds, and a maximum of 10 tokens are allowed in the token bucket. If the token bucket is full, a new token cannot be added until an existing token is used to enable sending an ICMP message. You can increase or decrease both the the frequency with which used tokens can be replaced and (optionally) the number of tokens allowed to exist.

error-interval: *Specifies the time interval in milliseconds between successive token adds. Increasing this value decreases the rate at which tokens can be added. A setting of* **0** *disables ICMP messaging.*
Default: *100;* ***Range:*** *0 - 2147483647.*

bucket-size: *This optional keyword specifies the maximum number of tokens allowed in the token bucket at any time. Decreasing this value decreases the maximum number of tokens that may be available at any time.*
Default: *10;* ***Range:*** *1 - 200.*

You can change the rate at which ICMP messages are allowed by changing the error-interval with or without a corresponding change in the bucket-size.

The **no ipv6 icmp error-interval** *command resets both the* **error-interval** *and the* **bucket-size** *values to their defaults.*

Use the **show run** *command to view the current ICMP error interval settings.*

For example, the following command limits ICMP error and informational messages to no more than 20 every 1 second:

```
ProCurve(config)# ipv6 icmp error-interval 1000000 bucket-size
20
```

Ping for IPv6 (Ping6)

The Ping6 test is a point-to-point test that accepts an IPv6 address or IPv6 host name to see if an IPv6 switch is communicating properly with another device on the same or another IPv6 network. A ping test checks the path between the switch and another device by sending IP packets (ICMP Echo Requests).

To use a **ping6** command with an IPv6 host name or fully qualified domain names, refer to "DNS Resolver for IPv6" on page 9-9.

You can issue single or multiple ping tests with varying repetitions and timeout periods to wait for a ping reply.

Replies to each ping test are displayed on the console screen. To stop a ping test before it finishes, press **[Ctrl] [C]**.

For more information about using a ping test, refer to the "Troubleshooting" appendix in the current *Management and Configuration Guide* for your switch.

Syntax: ping6 < *ipv6-address* | *hostname* | *switch-number* >
[repetitions < 1 - 10000 >] [timeout < 1 - 60 >] [data-size < 0 - 65507 >]
[data-fill < 0 - 1024 >] [source < *ipv6-addr* | *vid* >]
ping6 <*link-local-address*%vlan<*vid*> | *hostname* | *switch-number*>
[repetitions < 1 - 10000 >] [timeout < 1 - 60 >] [data-size < 0 - 65507 >]
[data-fill < 0 - 1024 >] [source < *ipv6-addr* | *vid* >]

Pings the specified IPv6 host by sending ICMP version 6 (ICMPv6) echo request packets to the specified host.

<*ipv6-address*>: *IPv6 address of a destination host device.*

< *link-local-address* >%vlan<*vlan-id*>: *IPv6 link-local address, where* **%vlan**<*vlan-id*> *specifies the VLAN ID number.*

< *hostname* >: *Host name of an IPv6 host device configured on an IPv6 DNS server.*

< *switch-number* >: *Number of an IPv6-based switch that is a member of a switch stack (IPv6 subnet). Valid values: 1 - 16.*

[repetitions < 1 - 10000>]: *Number of times that IPv6 ping packets are sent to the destination IPv6 host. Default: 1.*

[timeout <1 - 60>]: *Number of seconds within which a response is required from the destination host before the ping test times out. Valid values: 1 - 60. Default: 1 second.*

[data-size <0 - 65471]: *Size of data (in bytes) to be sent in ping packets. Valid values: 0 - 65471. Default: 0.*

[data-fill <0 - 1024>]: *Text string used as data in ping packets. Range: up to 1024 alphanumeric characters; Default: 0.*

[source < *ipv6-addr* | *vid* >]: *The IPv6 address of the pinging device or the VLAN-ID on which the ping is being sent.*

Default: **0** *(no text is used).*

```
ProCurve# ping6 fe80::2:1%vlan10
fe80:0000:0000:0000:0000:0000:0002:0001 is alive, time = 975 ms

ProCurve# ping6 2001:db8::a:1c:e3:3 repetitions 3
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 1, time = 15 ms
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 2, time = 15 ms
2001:0db8:0000:0000:000a:001c:00e3:0003 is alive, iteration 3, time = 15 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip (ms) min/avg/max = 15/15/15

ProCurve# ping6 2001:db8::214:c2ff:fe4c:e480 repetitions 3 timeout 2
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 1, time = 15 ms
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 2, time = 10 ms
2001:db8:0000:0000:0214:c2ff:fe4c:e480 is alive, iteration 3, time = 15 ms

ProCurve# ping6 2001:db8::10
Request timed out.
```

Figure 9-1. Examples of IPv6 Ping Tests

Traceroute for IPv6

The **traceroute6** command enables you to trace the route from a switch to a host device that is identified by an IPv6 address or IPv6 host name. In the command output, information on each (router) hop between the switch and the destination IPv6 address is displayed.

To use a **traceroute6** command with an IPv6 host name or fully qualified domain names, refer to "DNS Resolver for IPv6" on page 9-9.

Note that each time you perform a traceroute operation, the **traceroute** command uses the default settings unless you enter different values with each instance of the command.

Replies to each traceroute operation are displayed on the console screen. To stop a traceroute operation before it finishes, press **[Ctrl] [C]**.

For more information about how to configure and use a traceroute operation, refer to the "Troubleshooting" appendix in the *Management and Configuration Guide*.

Syntax: traceroute6 < *ipv6-address* | *hostname* >
[minttl < 1-255 > [maxttl < 1-255 >
[timeout < 1 - 60 >] [probes < 1-5 >] [source < *ipv6-addr* | *vid* >

traceroute6 <*link-local-address*%vlan<*vid*> | *hostname* >
[minttl < 1-255 >] [maxttl < 1-255 >]
[timeout < 1 - 60 >] [probes < 1-5 >] [source < *ipv6-addr* | *vid* >

Lists the IPv6 address of each hop in the route to the specified destination host device with the time (in microseconds) required for a packet reply to be received from each next-hop device.

<*ipv6-address*>: *IPv6 address of a destination host device.*

<*link-local-address*>%vlan<*vlan-id*>: *IPv6 link-local address, where* **%vlan<*vlan-id*>** *specifies the VLAN ID number.*

<*hostname*>: *Host name of an IPv6 host device configured on an IPv6 DNS server.*

minttl: *Minimum number of hops allowed for each probe packet sent along the route.* ***Default:*** *1;* ***Range:*** *1 - 255.*

- *If the* **minttl** *value is greater than the actual number of hops, the traceroute output displays only the hops equal to or greater than the configured* **minttl** *threshold value. The hops below the threshold value are not displayed.*
- *If the* **minttl** *value is the same as the actual number of hops, only the final hop is displayed in the command output.*
- *If the* **minttl** *value is less than the actual number of hops, all hops to the destination host are displayed.*

maxttl: *Maximum number of hops allowed for each probe packet sent along the route. Valid values: 1 - 255. Default: 30.*

- *If the* **maxttl** *value is less than the actual number of hops required to reach the host, the traceroute output displays only the IPv6 addresses of the hops detected by the configured* **maxttl** *value.*

timeout: *Number of seconds within which a response is required from the IPv6 device at each hop in the route to the destination host before the traceroute operation times out.* ***Default:*** *5 seconds;* ***Range:*** *1 - 60.*

probes: *Number of times a traceroute is performed to locate the IPv6 device at any hop in the route to the specified host before the operation times out.* ***Default:*** *3;* ***Range:*** *1 - 5.*

[source < *ipv6-addr* | *vid* >]: *The source IPv6 address or VLAN of the traceroute device or the VLAN-ID on which the traceroute packet is being sent.*

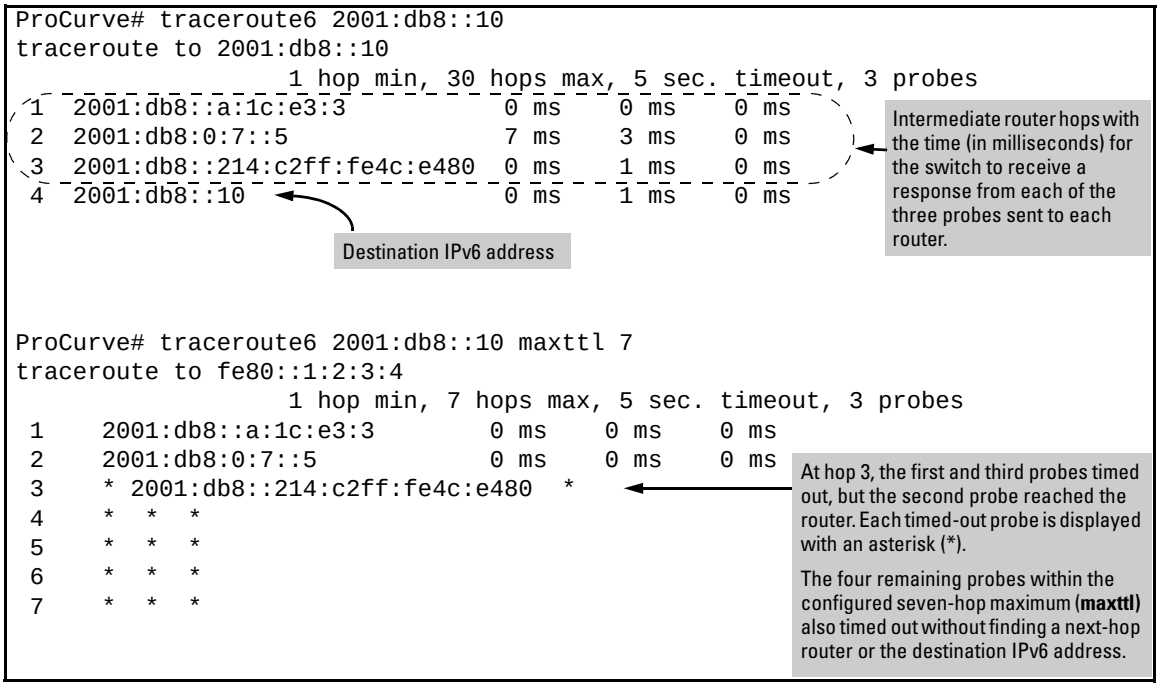
ProCurve# traceroute6 2001:db8::10
traceroute to 2001:db8::10
1 hop min, 30 hops max, 5 sec. timeout, 3 probes
1 2001:db8::a:1c:e3:3 0 ms 0 ms 0 ms
2 2001:db8:0:7::5 7 ms 3 ms 0 ms
3 2001:db8::214:c2ff:fe4c:e480 0 ms 1 ms 0 ms
4 2001:db8::10 0 ms 1 ms 0 ms
Intermediate router hops with the time (in milliseconds) for the switch to receive a response from each of the three probes sent to each router.
Destination IPv6 address
ProCurve# traceroute6 2001:db8::10 maxttl 7
traceroute to fe80::1:2:3:4
1 hop min, 7 hops max, 5 sec. timeout, 3 probes
1 2001:db8::a:1c:e3:3 0 ms 0 ms 0 ms
2 2001:db8:0:7::5 0 ms 0 ms 0 ms
3 * 2001:db8::214:c2ff:fe4c:e480 *
4 * * *
5 * * *
6 * * *
7 * * *
At hop 3, the first and third probes timed out, but the second probe reached the router. Each timed-out probe is displayed with an asterisk (*).
The four remaining probes within the configured seven-hop maximum (maxttl) also timed out without finding a next-hop router or the destination IPv6 address.

Figure 9-2. Examples of IPv6 Traceroute Probes

DNS Resolver for IPv6

The Domain Name System (DNS) resolver is designed for local network domains where it enables use of a host name or fully qualified domain name to support DNS-compatible commands from the switch. Beginning with software release K.13.01,DNS operation supports these features:

- dual-stack operation: IPv6 and IPv4 DNS resolution
- DNS-compatible commands: **ping**, **ping6**, **traceroute**, and **traceroute6**
- multiple, prioritized DNS servers (IPv4 and IPv6)

DNS Configuration

Up to three DNS servers can be configured. The addresses must be prioritized, and can be for any combination of IPv4 and IPv6 DNS servers.

Note

This section describes the commands for configuring DNS operation for IPv6 DNS applications. For further information and examples on using the DNS feature, refer to “DNS Resolver” in appendix C, “Troubleshooting”, in the current *Management and Configuration Guide* for your switch.

Syntax:. [no] ip dns server-address priority < 1 - 3 > < *ip-addr* >

Used at the global config level to configure the address and priority of a DNS server. Allows for configuring up to three servers providing DNS service. (The servers must all be accessible to the switch.) The command allows both IPv4 and IPv6 servers in any combination and any order of priority.

priority < 1 - 3 >: *Identifies the order in which the specified DNS server will be accessed by a DNS resolution attempt. A resolution attempt tries each configured DNS server address, in ascending order of priority, until the attempt is successful or all configured server options have been tried and failed. To change the priority of an existing server option, you must remove the option from the switch configuration and re-enter it with the new priority. If another server address is configured for the new priority, you must also remove that address from the configuration before re-assigning its priority to another address.*

— Continued on the next page. —

— Continued from the previous page. —

The **no** *form of the command removes the specified address from the server address list configured on the switch.*

< *ip-addr*>: *Specifies the address of an IPv6 or IPv4 DNS server.*

Syntax:. [no] ip dns domain-name < *domain-name-suffix* >

Used at the global config level to configure the domain suffix *that is automatically appended to the host name entered with a command supporting DNS operation. Configuring the domain suffix is optional if you plan to use fully qualified domain names in all cases instead of just entering host names.*

You can configure up to three addresses for DNS servers in the same or different domains. However, you can configure only one domain name suffix. This means that a fully qualified domain name must be used to resolve addresses for hosts that do not reside in the same domain as the one you configure with this command. That is, if the domain name suffix and the address of a DNS server for that same domain are both configured on the switch, then you need to enter only the host name of the desired target when executing a command that supports DNS operation. But if the DNS server used to resolve the host name for the desired target is in a different domain than the domain configured with this command, then you need to enter the fully qualified domain name for the target.

The **no** *form of the command removes the configured domain name suffix.*

For example, suppose you want to configure the following on the switch:

- the address **2001:db8::127:10** which identifies a DNS server in the domain named mygroup.procurve.net
- a priority of 1 for the above server
- the domain suffix **mygroup.procurve.net**

Assume that the above, configured DNS server supports an IPv6 device having a host name of “mars-1” (and an IPv6 address of fe80::215:60ff:fe7a:adc0) in the “mygroup.procurve.net” domain. In this case you can use the device's host name alone to ping the device because the mygroup.procurve.net domain has

been configured as the domain name on the switch and the address of a DNS server residing in that domain is also configured on the switch. The commands for these steps are as follows:

```
ProCurve(config)# ip dns server priority 1 2001:db8::127:10
ProCurve(config)# ip dns domain-name mygroup.procurve.net
ProCurve(config)# ping6 mars-1
fe80::215:60ff:fe7a:adc0 is alive, time = 1 ms
```

Figure 9-1. Example of Configuring for a Local DNS Server and Pinging a Registered Device

However, for the same "mars-1" device, if mygroup.procurve.net was not the configured domain name, you would have to use the fully qualified domain name for the device named mars-1:

```
ProCurve# ping6 mars-1.mygroup.procurve.net
```

For further information and examples on using the DNS feature, refer to "DNS Resolver" in appendix C, "Troubleshooting", in the current *Management and Configuration Guide* for your switch.

Viewing the Current Configuration

Use the **show ip dns** command to view the current DNS server configuration.

Use the **show run** command to view both the current DNS server addresses and the current DNS domain name in the active configuration.

Operating Notes

In software release K.13.01, DNS addressing is not configurable from a DHCPv6 server.

Debug/Syslog for IPv6

The Debug/System logging (*Syslog*) for IPv6 feature provides the same logging functions as the IPv4 version, allowing you to record IPv4 and IPv6 Event Log and debug messages on a remote device to troubleshoot switch or network operation. For example, you can send messages about routing misconfigurations and other network protocol details to an external device, and later use them to debug network-level problems.

Configuring Debug and Event Log Messaging

To specify the types of debug and Event Log messages that you want to send to an external device:

- Use the **debug** < *debug-type* > command to send messaging reports for the following types of switch events:
  - ACL “deny” matches
  - DHCP snooping events
  - Dynamic ARP protection events
  - Events recorded in the switch’s Event Log
  - IPv4 OSPF and RIP routing events
  - IPv6 DHCPv6 client and Neighbor Discovery events
  - LLDP events
- Use the **logging** < **severity** *severity-level* | **system-module** *system-module*> command to select a subset of Event Log messages to send to an external device for debugging purposes according to:
  - Severity level
  - System module

Debug Command

Syntax: [no] debug < *debug-type* >

Configures the types of IPv4 and IPv6 messages that are sent to Syslog servers or other debug destinations, where <*debug-type*> *is any of the following event types:*

acl

When a match occurs on an ACL "deny" statement with a **log** *parameter, an ACL message is sent to configured debug destinations. (Default: Disabled - ACL messages for traffic that matches "deny" entries are not sent.)*

all

Configures all IPv4 and IPv6 debug message types to be sent to configured debug destinations. (Default: Disabled - No debug messages are sent.)

arp-protect

Configures messages for Dynamic ARP Protection events to be sent to configured debug destinations. (Default: Disabled - No debug messages are sent.)

event

Configures Event Log messages to be sent to configured debug destinations.

Event Log messages are enabled to be automatically sent to debug destinations in the following conditions:

- *If no Syslog server address is configured and you enter the* **logging** *command to configure a destination address.*
- *If at least one Syslog server address is configured in the startup configuration and the switch is rebooted or reset.*

Event log messages are the default type of debug message sent to configured debug destinations.

ip

Configures IPv4 OSPF and RIP routing messages to be sent to configured debug destinations.

Syntax:. [no] debug < *debug-type* > *(Continued)*

ip [ospf < adj | event | flood | lsa-generation | packet | retransmission | spf >]

Configures specified IPv4 OSPF message types to be sent to configured debug destinations:
adj — *Adjacency changes.*
event — *OSPF events.*
flood — *Information on flood messages.*
lsa-generation — *New LSAs added to database.*
packet — *Packets sent/received.*
retransmission — *Retransmission timer messages.*
spf — *Path recalculation messages*

ip [rip < database | event | trigger >]

Configures specified IPv4 RIP message types to be sent to configured debug destinations:
database— *Database changes*
event— *RIP events*
trigger— *Trigger messages*

ipv6

Configures messages for IPv6 DHCPv6 client and neighbor discovery events to be sent to configured debug destinations.

ipv6 [dhcpv6-client <events | packets> | nd]

Configures one of the following IPv6 message types to be sent to configured debug destinations:
dhcpv6-clients events — *DHCPv6 client events*
dhcpv6-clients packets — *Statistics on DHCPv6 packets transmitted on a switch configured as a DHCPv6 client*
nd— *Events during IPv6 neighbor discovery*

lldp

Configures all LLDP message types to be sent to configured debug destinations.

wireless-services

Configures messages about the operation of wireless-services modules to be sent to configured debug destinations.

Configuring Debug Destinations

A Debug/Syslog destination device can be a Syslog server (up to six maximum) and/or a console session:

- Use the **debug destination** < **logging** | **session** | **buffer** > command to enable (and disable) Syslog messaging on a Syslog server or to a CLI session for the debug message types configured with the **debug** and **logging** commands (see "Configuring Debug and Event Log Messaging" on page 9-12):
  - **debug destination logging** enables the configured debug message types to be sent to Syslog servers configured with the **logging** command.
  - **debug destination session** enables the configured debug message types to be sent to the CLI session that executed this command. The session can be on any one terminal emulation device with serial, Telnet, or SSH access to the CLI at the Manager level prompt.
  - **debug destination buffer** enables the configured debug message types to be sent to a buffer in switch memory.

Logging Command

Syntax: [no] logging < *syslog-ipv4-addr* >

Enables or disables Syslog messaging to the specified IPv4 address. You can configure up to six addresses. If you configure an address when none are already configured, this command enables destination logging (Syslog) and the Event debug type. Therefore, at a minimum, the switch begins sending Event Log messages to configured Syslog servers. If other debug message types are configured, they are also sent to the Syslog server.

no logging *removes all currently configured Syslog logging destinations from the running configuration.*

no logging < *syslog-ipv4-address* > *removes only the specified Syslog logging destination from the running configuration.*

Note: *The* **no logging** *command does not delete the Syslog server addresses stored in the startup configuration. To delete Syslog addresses in the startup configuration, you must enter the* **no logging** *command followed by the* **write memory** *command. To verify the deletion of a Syslog server address, display the startup configuration by entering the* **show config** *command.*

To block the messages sent to configured Syslog servers from the currently configured debug message type, enter the **no debug** < *debug-type* > *command.*

To disable Syslog logging on the switch without deleting configured server addresses, enter the **no debug destination logging** *command.*

For complete information on how to configure a Syslog server and Debug/Syslog message reports, refer to the "Troubleshooting" appendix in the *Management and Configuration Guide.*

IPv6 Terminology

For IPv6 ACL terminology, refer to “Terminology” on page 8-9.

DAD	Duplicate Address Detection. Refer to “Duplicate Address Detection (DAD)” on page 4-18.
Device Identifier	The low-order bits in an IPv6 address that identify a specific device. For example, in the link-local address 2001:db8:a10:101:212:79ff:fe88:a100/64, the bits forming 212:79ff:fe88:a100 comprise the device identifier.
DoS	Denial-of-Service.
EUI-64	Extended Unique Identifier. Refer to “Extended Unique Identifier (EUI)” on page 3-14.
Manual Address Configuration	Configures an IPv6 address by using the CLI to manually enter a static address. Referred to as “Static Address Configuration” in this guide. See **Static Address Configuration**, below.
MLD	Multicast Listener Discovery. Refer to the chapter titled “Multicast Listener Discovery (MLD) Snooping”.
MTU	Maximum Transmission Unit. The largest frame size allowed on a given path or device. Refer to “Path MTU (PMTU) Discovery” on page 2-11.
RA	Router Advertisement. Refer to “Router Advertisements” on page 4-29.
SLAAC	Stateless Address Autoconfiguration. Refer to “SLAAC (Stateless Automatic Address Configuration)” on page 2-7.
Static Address	A permanently configured IPv6 address, as opposed to an autoconfigured address.
Static Address Configuration	Configures an IPv6 address by using the CLI to manually enter the address instead of using an automatically generated or DHCPv6-assigned address. Same as “Manual Address Configuration”. See also **Manual Address Configuration**, above.

Index

Symbols

… 4-7, 4-13
%vlan suffix … 5-6, 5-10, 5-13

Numerics

802.1X
ACL, IPv6, effect on … 8-17
port-based access not recommended … 8-17

A

ACL
debug messages … 9-13
end … 8-41
filtering process … 8-30
rules, operation … 8-31
traffic not filtered … 8-30
VLANs … 8-32
ACL, IPv4
802.1X port-based not recommended … 8-17
deny any, implicit, IPv6 … 8-16, 8-4
limit … 8-42, 8-47
monitoring … 8-96
RADIUS-assigned, limit … 8-42, 8-47
scalability … 8-42, 8-47, 8-96, 8-101
ACL, IPv6
802.1X client limit … 8-17
ACE
after match not used … 8-30, 8-40
defined … 8-9
general rules … 8-43
insert in list … 8-68
limit … 8-31
minimum number … 8-105
not used … 8-26
See sequence, ACEs.
address family … 8-10
AppleTalk … 8-30
application … 8-4, 8-29, 8-22, 8-29, 8-6, 8-13, 8-18, 8-22, 8-24, 8-35, 8-42, 8-35, 8-62, 8-63, 8-65
basic structure … 8-36
CIDR … 8-10
mask … 8-43
command summary … 8-8, 8-46, 8-13, 8-22, 8-45, 8-42, 8-35, 8-21, 8-57, 8-88, 8-7, 8-45, 8-42
DA, defined … 8-10, 8-12
defined … 8-4, 8-9, 8-36, 8-9, 8-8, 8-46, 8-65, 8-37, 8-21, 8-24, 8-27, 8-29, 8-31, 8-37, 8-40, 8-41, 8-66, 8-11, 8-10
ACLs and assignments … 8-86
assignments … 8-81, 8-82
configuration details … 8-80, 8-83
data types … 8-86
summary, configured ACLs … 8-79
DSCP setting … 8-22
dual stack … 8-18, 8-4, 8-43, 8-4, 8-6, 8-24, 8-4, 8-12, 8-16, 8-14, 8-32
editing … 8-66, 8-87, 8-42, 8-62, 8-10, 8-66, 8-36, 8-57, 8-41
features, common to all … 8-21, 8-13, 8-26
hit count
See statistics, ACE.
ICMP
options … 8-58
traffic … 8-22, 8-59
… 8-10
identifier … 8-11, 8-62, 8-64
See deny any, implicit.
… 8-11, 8-24
See ACL, IPv6, wildcard.
IPX … 8-30
length, prefix … 8-21, 8-42, 8-47, 8-21
See ACL, IPv6, logging.
… 8-21, 8-22
described … 8-92
session … 8-21
… 8-105
mask
CIDR … 8-43
defined … 8-10
… 8-41, 8-30, 8-31, 8-96, 8-18, 8-20, 8-18
name or number assignment … 8-42, 8-11, 8-36, 8-62, 8-64, 8-42, 8-30, 8-21
offline editing … 8-87, 8-55, 8-56, 8-11
packet screened by multiple lists … 8-20, 8-12, 8-31, 8-12, 8-18, 8-36, 8-22, 8-28, 8-5, 8-32, 8-6, 8-13

See also static port ACL and dynamic port ACL.
… 8-32, 8-17, 8-19, 8-32, 8-22, 8-53
ACL, IPv6
mask, IPv4 … 8-10
… 8-21, 8-36, 8-5
RACL
operation defined … 8-13
… 8-4, 8-6, 8-9, 8-10, 8-16, 8-24, 8-9
implicit deny IPv6 … 8-16
multiple clients connected … 8-16
denied traffic … 8-16
… 8-42, 8-47
remark … 8-12
remove from an ACE … 8-75
… 8-62, 8-63, 8-32, 8-42, 8-7, 8-45, 8-106, 8-31
SA, defined … 8-12
scalability … 8-42, 8-47, 8-5, 8-29, 8-30, 8-7, 8-12, 8-45, 8-66
out-of-range … 8-68
use to delete ACE … 8-70, 8-68
… 8-43, 8-9, 8-6, 8-8, 8-13, 8-46, 8-16, 8-13, 8-32, 8-6, 8-96, 8-38, 8-40
Syslog
See ACL, logging.
TCP control bits … 8-7, 8-36, 8-38, 8-45, 8-47, 8-57, 8-7, 8-45, 8-56, 8-36, 8-56
terms … 8-9, 8-5, 8-28, 8-96, 8-32, 8-13, 8-32, 8-41, 8-78, 8-81, 8-82, 8-83
user-based 802.1X … 8-17, 8-19
VACL
configure … 8-8, 8-46
defined … 8-6
operation defined … 8-13
VACL applications … 8-15
See VACL.
wildcard … 8-10, 8-13
address configuration
DNS for IPv6 … 2-14
duplicate unicast addresses … 3-6, 2-9, 4-18
IPv6 anycast address … 2-9, 2-11, 2-7, 2-8, 3-5, 3-11, 3-16, 3-17, 4-7, 4-13, 2-8, 3-5, 3-6, 3-8, 4-9, 2-8, 3-5, 4-12, 2-7, 3-5, 3-11, 3-13, 4-6, 3-11
maximum number of IPv6 addresses … 2-16, 3-3, 3-5, 3-9
neighbor discovery for IPv6 … 2-15, 3-4
omitting zeros in IPv6 address … 3-3
single IPv6 local-link address on an interface … 3-13
See also IPv6.
address family … 8-10
all-nodes, used in IPv6 DAD … 4-18
anycast address … 5-2
DAD not supported … 3-20
deprecation … 4-34
in IPv6 … 2-9
IPv6 address … 3-10, 3-20, 4-14
preferred lifetime … 4-34
valid lifetime … 4-34
ARP protection
debug messages … 9-13
authorized IP managers
binary expressions of hexadecimal blocks … 6-7, 6-11
configuration command … 6-5, 6-8, 6-13, 6-4
displaying configuration … 6-12
feature description … 6-3
IP mask used to configure single station … 6-5, 6-6
precedence among security settings … 6-4
using IP masks … 6-3, 6-5
authorized ip managers
access privilege … 6-5
autoconfigured address
effect of static address … 4-14
autoconfigured unicast address
DHCPv6 precedence … 4-11
autorun
TFTP download of key file … 5-17
auto-TFTP
disabled … 5-20
for IPv6 … 5-20

B

binary expressions of IPv6 address … 6-7, 6-11

C

clear neighbor cache … 5-2, 5-5
command file
TFTP download and running command script … 5-17
command index, IPv6 … -xv
command output

TFTP upload on remote device … 5-18
command prompts … 1-3
command syntax conventions … 1-2
configuration file
TFTP download … 5-17, 5-18
control bits, TCP … 8-57
copy
TFTP transfers … 5-15
crash data file
TFTP upload on remote device … 5-18
crash log
TFTP upload on remote device … 5-18

D

DA, defined … 8-10, 8-12
DAD
configuration … 4-19
detecting duplicate unicast addresses … 3-6, 4-18, 2-9, 4-5, 4-8, 4-10, 4-12, 4-16
not supported on anycast addresses … 3-20
performed on all IPv6 unicast addresses … 4-20
debug
compared to event log … 9-12
forIPv6 … 9-12
sending event log messages … 9-12
using CLI session … 9-15
debug command
DHPv6 messages … 9-14
event log messages … 9-13
IPv4/IPv6 event messages … 9-13, 9-12
LLDP messages … 9-14
OSPF messages … 9-14
RIP messages … 9-14
using Syslog servers … 9-15
wireless-services messages … 9-14
default settings
IPv6
access-list resequence interval, *10* … 8-71, 4-8, 4-3
DAD, *enabled* … 4-22
dhcp rapid-commit, *disabled* … 4-10
ICMPv6 error message generation, *enabled* … 9-2, 9-3, 4-3
managemenent features … 5-2, 6-2
MLD default mode, *auto* … 7-5, 2-11
nd dad-attempts, *3 (enabled)* … 4-19, 4-20
ping6 data-size and data-fill, *0* … 9-5, 9-4, 9-5
SSHv2, *enabled* … 2-12
traceroute 6 defaults … 9-7
denial-of-service
ICMPv6 rate limiting … 2-14
deprecated address … 4-23
device identifier in IPv6 address … 3-4
See also interface identifier.
DHCPv6
debug messages … 9-14
DHCP relay for IPv6 … 3-8
does not assign link-local address … 4-9, 3-8
mutually exclusive with autoconfigured global unicast address … 4-7, 4-11
NTP server … 2-8
precedence over autoconfig address … 4-11
server-assigned global unicast address … 2-8, 3-5, 3-6, 3-8, 4-9, 4-10
timep server … 2-8
DNS
configuration … 9-9
domain-name … 9-10
for IPv6 … 2-14
view configuration … 9-11
documentation
installation guide … 1-9
latest versions … 1-2, 1-4, 1-7
sources for more information … 1-4
dual-stack operation … 2-6, 8-4
switching IPv4 and IPv6 traffic on same VLAN … 2-3, 2-4, 3-6
using DHCPv6 … 3-8
duplicate address detection
See DAD.

E

EUI
in IPv6 address autoconfiguration … 4-7, 4-13
used in IPv6 address autoconfiguration … 2-7, 3-4, 3-5, 3-13, 3-14, 4-6
event log
compared to debug/Syslog operation … 9-12
debug messages … 9-13, 9-12
IPv6 support … 2-15
TFTP upload on remote device … 5-18
extended unique identifier
See EUI.

F

fast leave
MLD configuration … 7-10, 7-11
used in MLD snooping … 7-7
FD, unique local unicast address prefix … 3-12, 3-19
FE80
link-local address prefix … 3-11, 4-6
FE80, link-local address autoconfiguration … 2-7, 3-9, 3-13, 3-14
FF, IPv6 multicast address prefix … 3-12
flow sampling … 5-21

G

gateway
determining default IPv6 route … 2-8, 4-31
global unicast address
autoconfiguration … 3-5, 3-11, 3-16, 4-7
default prefix … 3-18, 3-16, 4-34, 3-18
leading 2 in prefix … 3-12
manual configuration … 2-8, 3-5, 3-9, 3-17, 4-13
network prefix … 3-4
preferred lifetime … 3-25, 4-8, 4-10, 4-12, 4-34
valid lifetime … 3-25, 4-8, 4-10, 4-34

I

IANA … 8-56
IANA, protocol numbers … 8-51, 8-58
ICMP
bucket-size … 9-3
error-interval … 9-3
for IPv6 … 2-14
rate-limiting controls … 9-2
inform messages … 5-21
interface identifier
in global unicast address … 3-18
IP authorized managers
for IPv6 … 2-12
IP masks
for multiple authorized manager stations … 6-6, 6-5
used in configuring authorized IP management … 6-5, 6-3
IP Preserve
configuring … 5-24
DHCP-assigned address … 5-25
downloading configuration file to IPv6 switch … 5-25
feature description … 5-24, 2-11
IPv6
address format … 3-3, 2-9, 3-10, 3-20, 4-14, 5-2
benefits … 2-6
command index … -xv, 4-4
DAD … 4-18
debug … 9-12, 2-8, 4-31
DHCPv6 server-assigned address … 2-8, 3-5, 3-6, 3-8, 4-4, 4-9
disabling … 4-16, 4-22, 4-27, 4-31, 4-32
DNS configuration … 9-9, 2-14
dual-stack operation … 2-3, 2-4
enabling commands … 3-14, 4-5
displayed in IPv6 configuration … 4-27
… 2-15
global unicast address autoconfiguration … 2-7, 3-5, 3-11, 3-16, 4-7, 3-16, 3-25, 2-8, 3-5, 3-9, 3-17, 4-13
ICMP error messages … 2-14, 2-12, 2-11, 5-24
link-local address autoconfiguration … 2-7, 3-5, 3-11, 3-13, 4-6, 2-8, 3-5, 3-9, 4-12, 5-6, 5-10, 5-13, 2-15, 3-24
management station … 2-7, 2-3, 2-4
MTU … 2-9, 2-11
multicast … 2-9, 2-6, 3-10, 3-21, 3-22
See MLD.
… 3-3, 3-5
neighbor cache, clear … 5-5, 5-3, 2-9, 2-15, 4-17, 5-2, 3-4
omitting zeros in address … 3-3
ping6 … 2-11, 2-14, 3-6
restrictions … 2-16, 4-29
security features … 2-12, 4-30, 3-13
SNMP support … 2-15, 5-21
See SNTP server.
… 2-12
See also SSH.
static address configuration … 4-11, 1-2, 2-3
Syslog … 9-12
Telnet … 2-10, 5-7
telnet6 … 5-6
Telnet6, access … 5-8, 2-10, 5-15
time protocols … 2-8, 2-10
Timep
See Timepv6.
traceroute6 … 2-14

for IPv6 … 2-14
… 2-5
unicast address … 3-10, 3-11, 3-19, 3-25, 2-6, 2-4
web browser interface … 2-11, 3-7
See also MLD.
IPv6 address
binary expression … 6-7, 6-11
ipv6 enable … 3-14, 4-5, 4-6
IPv6 interface identifier

L

link-local address
autoconfiguration … 2-7, 3-5, 3-11, 3-13, 4-6, 3-14
manual configuration … 2-8, 3-5, 3-9, 4-12
network prefix … 3-4
one address per interface … 3-13
LLDP
debug messages … 9-14
local unicast address
network prefix … 3-4
logging command
configuring a Syslog server … 9-16
syntax … 9-12
loopback address … 2-15, 3-24

M

MAC address
used in IPv6 interface identifier … 3-4, 4-6, 2-7, 3-5, 3-13, 3-14, 4-6
manual address configuration
See static address configuration.
masks
See IP masks.
maximum transmission unit … 2-9
See MTU.
MIB support
SNMP … 5-21
migration from IPv4 to IPv6 … 2-3, 2-4, 2-6
mirroring
ACL, classifier-based … 8-20
MLD
blocking multicast packet forwarding … 7-5, 7-9
configuration … 7-8
displaying configuration … 7-12, 7-15, 7-18, 7-20
forwarding multicast packets … 7-5, 7-9
overview … 2-11
reducing multicast flooding … 7-2, 7-4
snooping at port level … 7-2
used on IPv6 local link … 7-2
MTU
for IPv6 … 2-11, 2-9
multicast
IPv6 address … 2-6, 3-10, 3-21, 3-22, 3-4, 3-12, 3-21, 3-23, 2-9
MLD snooping reduces multicast flooding … 7-2, 7-4
Multicast Listener Discovery
See MLD.

N

neighbor cache, view … 5-3
neighbor discovery
for IPv6 nodes … 2-15
IPv6 similar to IPv4 ARP … 2-9, 4-17
neighbor solicitations
used in duplicate address detection … 4-19
neighbor, clear cache … 5-2
notifications
displaying configuration … 5-23
supported in IPv6 … 5-21
NTP server … 2-8

O

OSPF
debug messages … 9-14
outbound Telnet6 … 5-6

P

ping6 … 2-14, 9-4
ping6 on web browser … 2-11
port
MLD snooping … 7-17
port ACL … 8-6
port-level MLD snooping … 7-2, 7-9
preferred address … 4-23
preferred lifetime … 4-23
of global unicast address … 3-7, 3-25, 4-8, 4-10, 4-12
use of IPv6 address as source or destination … 4-34
priority

public-key file
TFTP download … 5-18

R

RADIUS
dynamic port ACL
See also RADIUS-assigned ACLs.
RADIUS-assigned ACLs … 8-16
RADIUS-assigned
See also dynamic port acl
RADIUS-assigned ACLs … 8-6, 8-16
rate-limiting
ACL, static, classifier-based … 8-20
resource monitor
See *Management and Configuration Guide.*
RIP
debug messages … 9-14
router advertisements
used in IPv6 … 4-29
routing
determining an IPv6 gateway … 2-8
DHCPv6 debug messages … 9-14, 2-8, 3-5, 3-6, 3-8, 4-9
displaying IPv6 routing table … 4-31, 4-32, 2-6
IPv6 global unicast address autoconfiguration … 2-7, 3-5, 3-11, 3-16, 4-7, 4-30, 3-16, 3-25, 4-25, 4-29, 2-5, 3-19, 3-11
maximum number of IPv6 routes … 2-16
OSPF debug messages … 9-14
RIP debug messages … 9-14
selecting default IPv6 router … 4-30, 2-4
traceroute … 9-6
running-config
TFTP upload on remote device … 5-18

S

SA … 8-12
SCP
See SCP/SFTP.
SCP/SFTP
secure file transfer … 6-19
secure copy
See SCP/SFTP.
secure FTP
See SCP/SFTP.
security
for IPv6 … 2-12
IPv6 authorized managers … 2-12
precedence of authorized IP manager settings … 6-4
SSHv2 for IPv6 … 2-12
security, ACL, IPv6
See ACL, IPv6, security use.
sFlow … 5-21
SFTP
See SCP/SFTP.
show ipv6 … 2-9, 3-6, 4-6, 4-8, 4-10, 4-13, 4-15, 4-22
show run
IPv6 output … 4-27
SNMP
configuring SNMPv1/v2c trap receiver … 5-22
displaying SNMPv3 management station configuration … 5-24, 5-23
features supported for IPv6 … 5-21
IPv6 support … 2-15
remote monitoring (RMON) … 5-21
SNMPv1 and v2c traps … 5-21
source IPv6 address in notifications not supported … 5-22, 5-21
SNTP
mode … 5-11
poll interval … 5-11
server address … 5-11
view configuration … 5-11
SNTP server … 5-13
address configuration
IPv6 address
priority
SNTPv6 … 2-10
software image
TFTP download … 5-18
solicited-node
IPv6 multicast address group … 3-21, 3-23
used in IPv6 neighbor discovery … 4-17
SSH
filetransfer … 5-20, 2-12
overview … 6-15
SSHv2 restriction … 6-18
version 1 … 6-18
startup-config
TFTP download … 5-18
stateless automatic address configuration … 2-7
static ACL … 8-6
static address configuration … 4-11

effect of autoconfig … 4-14
subnetting
in IPv6 … 3-3, 3-5, 3-9
suffix, link-local address … 5-6, 5-10, 5-13
supersede implicit deny any any … 8-37
Syslog
compared to event log … 9-12
event log messages sent by default … 9-16
for IPv6 … 9-12
See ACL, IPv6, logging.
sending event log messages … 9-12

T

TCP control bits … 8-57
Telnet
viewing current use … 5-7
Telnet6 … 5-6
enable/disable inbound … 5-8
operations supported … 2-10
view configuration … 5-8
TFTP
auto-TFTP feature … 5-20
disabled … 5-20, 5-17, 5-18, 5-17
enabling client functionality … 5-16
uploading command output … 5-18
TFTP6
auto-TFTP … 5-20
copy command … 5-15, 5-17
enable client or server … 5-16
file transfers over IPv6 … 5-15, 2-10
See also IPv6. … 5-15
upload file to server … 5-18
time sync mode … 5-11
timep server … 2-8
Timepv6 … 2-10, 5-13
manual configuration … 5-13
traceroute … 9-6
for IPv6 … 2-14
traceroute6 … 9-6
traffic monitoring
sFlow … 5-21
traps
displaying configuration … 5-23
supported in IPv6 … 5-21
troubleshooting
configuring Syslog servers … 9-15
IPv6 addresses in event log … 2-15
ping6 … 2-14
traceroute6 … 2-14
using CLI session … 9-15, 2-14, 2-15, 9-12
trunk
ACL static trunk assignment … 8-13
port added or removed, ACL … 8-32
tunneling … 2-5

U

unicast
IPv6 address … 3-10
unique local unicast address
autoconfiguration … 3-11
used within an organization … 3-19
unspecified address
in IPv6 … 3-25

V

VACL defined … 8-6
valid lifetime
of global unicast address … 3-7, 3-25, 4-8, 4-10
use of deprecated IPv6 address as source or destination … 4-34
VLAN
deprecated global unicast address … 3-16, 3-25
DHCPv6 server-assigned address … 4-9
displaying IPv6 configuration … 4-25, 4-27, 4-32, 7-12, 7-15, 7-17, 7-18, 7-20, 2-4, 2-6
global unicast address autoconfiguration … 2-7, 3-5, 3-11, 3-16, 4-7, 2-8, 3-5, 3-9, 3-17, 4-13, 3-12
IPv6 link-local address autoconfiguration … 4-6, 3-21
link-local address autoconfiguration … 2-7, 3-5, 3-13, 3-14, 4-6, 2-8, 3-5, 3-9, 4-12, 3-11
MLD snooping … 7-5, 7-8, 7-9, 7-10
neighbor discovery operation … 4-17
router advertisements used in IPv6 … 4-29
selecting default IPv6 router … 4-30, 2-3, 3-6, 2-3
unique local unicast address configuration … 3-11, 3-12, 2-4

W

warranty … -ii
web browser … 1-8
See also web browser interface.

web browser interface
IPv6 support … 2-11
wildcard
See ACL, IPv6, wildcard.
wireless services
debug messages … 9-14

Technology for better business outcomes

To learn more, visit www.hp.com/go/procurve/

© Copyright 2009 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP will not be liable for technical or editorial errors or omissions contained herein.

June 2009

Manual Part Number
5992-3067